

FOCUS OF ATTENTION BASED ON GAMMA KERNELS
FOR AUTOMATIC TARGET RECOGNITION

By
MUNCHURL KIM

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1996

UNIVERSITY OF FLORIDA LIBRARIES

To my mother,

*who loved and supported his son with a great deal of endurance
in spite of many difficulties during the period of his son's study.*

ACKNOWLEDGEMENTS

I would first like to thank my advisor, Dr. Jose C. Principe, for his patient guidance and inspiration during my study. His availability to his students and the abundance of the ideas he offered are irreplaceable.

Special thanks are due to Dr. John G. Harris for constructive comments to this study. I also thank my committee members Dr. William W. Edmonson, Dr. Jian Li, and Dr. Joseph N. Wilson for serving on my supervisory committee. I would like to thank John Fisher III for sharing his broad knowledge and large experience of radar with me and for helping me conduct ATD/R research in the Computational NeuroEngineering laboratory. A special thank is due to Frank Candocia for his careful proofreading of this dissertation.

I am especially grateful to my parents who supported and prayed for their son with a great deal of love. Finally, I would like to thank my wife, SoYoung Lee, and my son, Mujoon, for their love during the course of my studies.

TABLE OF CONTENT

	<u>page</u>
ACKNOWLEDGEMENTS	iii
ABSTRACT	viii
CHAPTERS	
1 INTRODUCTION	1
1.1 Automatic Target Detection/Recognition Technology	1
1.2 A Multi-stage Automatic Target Detection/Recognition System	2
1.3 Overview of the Dissertation	6
2 SYNTHETIC APERTURE RADAR (SAR) DATA DESCRIPTION	7
2.1 Introduction	7
2.2 SAR Image Formation	8
2.2.1 Range Processing	9
2.2.2 Azimuth Processing	13
2.3 ISAR Image Formation	16
2.4 Preprocessing	20
2.4.1 Polarimetric Clutter Model	20
2.4.2 Polarimetric Whitening Filter (PWF) Processing	23
2.4.3 Preprocessing SAR Imagery	26
2.5 SAR Image Visualization	28
2.6 Examples of SAR Images	30
2.7 Target Embedding Strategy	31
2.7.1 Development of a Target Embedding Method	31
2.7.2 Embedding the TABLIS 24 ISAR Target Data into the Mission 90 Pass 5 SAR Data Set	34
3 PRESCREENERS	38

3.1 Introduction	38
3.2 A One-parameter Constant False Alarm Rate (CFR) Detector	38
3.3 A Two-parameter CFAR Detector	40
3.4 Extension to the Two-Parameter CFAR Detector	43
3.4.1 Introduction	43
3.4.2 Gamma Filter and Gamma Kernels	43
3.4.3 2-D Extension of 1-D Gamma Kernels	49
3.4.4 Gamma CFAR (γ CFAR) Detector	50
3.4.5 Implementation of the γ CFAR Detector	56
3.5 Receiver Operating Characteristics (ROC)	57
4 QUADRATIC GAMMA DETECTOR	58
4.1 Introduction	58
4.2 Discriminant Functions	58
4.2.1 Linear Discriminant Functions	58
4.2.2 Generalized Linear Discriminant Functions	59
4.3 Extension to the γ CFAR detector	60
4.3.1 Training the QGD	63
4.3.1.1 Close Form Pseudo-Inverse Solution	63
4.3.1.2 Iterative Solution Based on Gradient Descent	64
4.3.2 1-D Implementation	69
4.4 Comparison of the QGD with the CFAR and the γ CFAR Detectors	70
4.5 Artificial Neural Networks	70
4.5.1 Introductions	70
4.5.2 Multi-layer Perceptrons (MLPs)	71
4.5.3 Training MLPs	73
4.5.3.1 On-Line Learning	73
4.5.3.2 Learning Rate and Momentum	76
4.5.3.3 Batch Learning	77
4.5.4 Validation of A Neural Model	78
4.5.4.1 Bias/Variance Dilemma	78
4.5.4.2 Network Complexity and Early Stopping	79
4.6 Nonlinear Extension to the QGD (NL-QGD)	80

4.6.1	Introduction	80
4.6.2	Training the NL-QGD	81
5	TRAINING STRATEGIES FOR NL-QGD	85
5.1	Introduction	85
5.2	Optimality Index	86
5.2.1	L_2 Norm	86
5.2.2	Training with Excluding Outliers from Non-Target Class with L_2 Norm	89
5.2.3	L_p Norm	92
5.2.4	Mixed L_p Norm	95
5.2.5	Cross Entropy	96
6	EXPERIMENTS AND RESULTS	100
6.1	Introduction	100
6.2	Prescreening SAR Imagery	100
6.2.1	Two-Parameter CFAR Processing	100
6.2.2	γ CFAR Processing	103
6.2.2.1	Optimal Parameter Search	103
6.2.2.2	Impact of Stencil Size in False Alarm	104
6.2.2.3	Batch-running the γ CFAR Detector	107
6.2.3	Performance Comparison of the Two-Parameter CFAR Detector and the γ CFAR Detector	107
6.2.4	Conclusion	109
6.3	False Alarm Reduction	114
6.3.1	CFAR/QGD	114
6.3.1.1	Training QGD	114
6.3.1.1.1	Training Data Preparation	114
6.3.1.1.2	Optimal Weights by the Closed Form Solution	114
6.3.1.1.3	Testing the QGD	118
6.3.1.2	Training the QGD in an Iterative Manner	118
6.3.1.3	Independent Testing Results for the QGD	125
6.3.1.4	Conclusion	127

6.3.2 CFAR/NL-QGDs	127
6.3.2.1 L_2 Based Training and Testing of the NL-QGDs	127
6.3.2.2 Training and Testing of the NL-QGDs without Non-Target Outliers	135
6.3.2.3 L_p Based Training and Testing of the NL-QGDs	140
6.3.2.4 Training and Testing of the NL-QGDs with Mixed L_p Norms	147
6.3.2.5 Training and Testing of the NL-QGDs with Cross Entropy	153
6.3.3 Summary of Detection Performance in the CFAR/QGD and CFAR/NL-QGDs	158
6.3.4 γ CFAR/QGD and γ CFAR/NL-QGDs	162
6.3.5 Fast Implementation of γ CFAR/QGD and γ CFAR/NL-QGDs ..	163
7 CONCLUSIONS AND FUTURE WORKS	167
7.1 Summary	167
7.2 Future Works	170
APPENDIX	171
POLARIZATION BASIS TRANSFORMATION	171
A.1 Representation of a Plane Wave Polarization	171
A.2 Circular-to-Linear and Linear-to-Circular Polarization Basis Transformations	173
REFERENCES	177
BIOGRAPHICAL SKETCH	184

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

FOCUS OF ATTENTION BASED ON GAMMA KERNELS
FOR AUTOMATIC TARGET RECOGNITION

By

MUNCHURL KIM

August 1996

Chairperson: Dr. Jose C. Principe
Major Department: Electrical and Computer Engineering

A multi-stage approach has been attractive to avoid prohibitive processing in automatic target detection/recognition (ATD/R) of sensor imagery. Multi-stage ATD/R systems usually implement a focus of attention stage to localize the regions of interest where possible target candidates are found and apply a recognition stage only at the regions of interest selected by the focus of attention stage. The focus of attention stage rejects most clutter (uninteresting background) in sensor imagery and detects targets of interest at a high probability of detection rate (usually 100% detection).

This dissertation addresses a novel approach to the design of a focus of attention stage in a multi-stage ATD/R system. The focus of attention stage consists of two subsystems: (1) a front-end detection stage in which a conventional two-parameter constant false alarm rate (CFAR) detector is extended to the gamma CFAR (γ CFAR) detector. The γ CFAR detector relaxes the constraint of a fixed stencil size in the two-parameter CFAR stencil by using gamma kernels; (2) a false alarm reduction stage implementing a quadratic detector of the intensity features estimated with the gamma kernels, which is called the quadratic

gamma detector (QGD). The QGD extends the two-parameter CFAR test with respect to: i) the stencil shape; ii) the features used in the decision function; iii) the selection of weights which are not a priori chosen but are found through optimization.

The QGD is further extended to a nonlinear adaptive structure (multi-layer perceptron) which is denoted by the NL-QGD. The training strategies of the NL-QGD are discussed in terms of detection theory. Several norms such as L_8 , $L_{1,1}/L_8$, cross entropy function, L_2 with removal of non-target outliers during the training are implemented to train the NL-QGD. The effect of different norms is measured in terms of receiver operating characteristic (ROC) in a large data set of synthetic aperture radar (SAR) clutter (about 7 km^2) with targets embedded. With these new criteria, the NL-QGD was able to surpass the performance of the QGD.

CHAPTER 1

INTRODUCTION

Automatic Target Detection/Recognition (ATD/R) is a challenging problem. The goal is to detect and recognize objects of interest in a clutter dominated imagery (e.g., a forward-looking infrared radar, synthetic aperture radar or laser radar etc.).

Early radar systems displayed all incoming information on a screen. Clutter, noise, and target amplitude variations were displayed simultaneously. Target detection was performed by human operators, visually monitoring image intensity variations in order to discriminate targets against background clutter and noise. These raw data displays are still incorporated, in some sense, into most major systems. The objective of automatic detection processing is to automatically detect targets and to provide target reports without human intervention.

Background clutter, which usually dominates sensor imagery, may be divided into two clutter types; natural clutter which describes natural scenery (trees, bushes, grass and forest etc.) and cultural clutter which envelops man-made objects (cars, bridges, power lines, buildings etc.). Background clutter may not be typically neither stationary, ergodic, nor Gaussian, especially in high resolution imagery [55] [87]. Target signatures can vary depending upon viewing angle and posture. The difficulty of the ATD/R problem ascribes to such complicated variations of target signatures and background clutter in sensor imagery.

1.1 Automatic Target Detection/Recognition Technology

Sensor technology and computing power have made great progress in forming and

acquiring sensor data. However, relatively less progress has been made in ATD/R algorithms. Many ATD/R approaches have been proposed which include detection theory [42], pattern recognition technique [9] [11] [47], neural networks [3] [11] [76] [82] [87] [88], and model-based algorithms [21] [84] [85].

Detection theory is attractive for the ATD/R problem. When target signatures and back-ground are described by statistical models an optimal detection can be theoretically derived, that is, a required detection probability can be determined given a false alarm rate. The advantage of the detection theory approach is that target signatures and background clutter can be expressed in an efficient way by statistical parameters and optimal solutions can be derived. However, this approach requires that the statistical model be valid and analytically tractable for target signatures and background clutter. When the statistical model does not adequately describe real-life raw data, it degrades detection performance.

Pattern recognition representations typically involve feature extraction from targets and background. The features are essential to the target recognition process. Distinction between different target types and background clutter should clearly be based on the extracted features.

While many such efforts have been made to solve the ATD/R problem, none so far has succeeded because variations in both target signature and background clutter contribute to the difficulty of the ATD/R problem [47].

1.2 A Multi-stage Automatic Target Detection/Recognition System

Besides the significant variation of target signatures and background clutter which adds to the difficulty of the ATD/R problem, ATD/R systems usually have to deal with prohibitive amounts of image data. Furthermore, it is attractive to seek the construction of a single algorithm which exploits all of the information of high resolution imagery and solve the ATR problem. The single-algorithm approach is computationally too expensive and high resolution imagery is difficult to model accurately and hence is poorly under-

stood. Due to the prohibitive amount of sensor imagery to be processed, real-time processing requirements mandate efficient algorithms and powerful processing architectures. The multistage approach becomes an attractive alternative because it progressively reduces the number of interesting areas of the image and narrows down their consideration in the further stages, allowing a recognition algorithm to avoid the processing of entire images [3] [11] [57] [58] [60] [76] [86]. Figure 1 shows a conceptual flow of image data processed in multistage ATR systems.

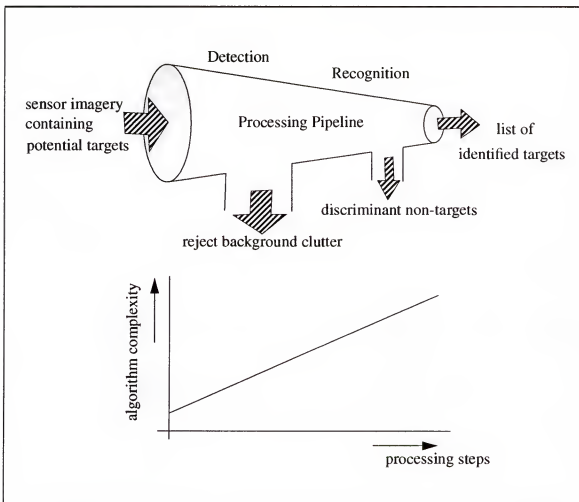


Figure 1 Data processing flow and algorithm complexity of a multi-stage ATR system¹.

1. The algorithm complexity does not necessarily increase linearly with the processing steps.

The detection stage can be thought of as a data reduction stage. A simple prescreening algorithm in the detection stage operates over the entire imagery and selects regions of interest (ROIs) where all target-like objects are found, and passes the locations of the ROIs to the recognition stage for further consideration. The recognition stage deploys a recognition algorithm only over the ROIs, rejects non-targets and recognizes the remaining targets.

The Lincoln Laboratory baseline ATR system [57] consists of three stages which include the prescreening stage (or front-end detection stage), the discrimination stage, and the classification. A two-parameter constant-false alarm (CFAR) detector serves as a prescreener in the front-end detection stage and locates all possible target-like objects based on pixel intensity. The discrimination stage receives the locations in which target candidates are found from the prescreener and rejects natural clutter [9] [57]. In the classification stage, the classifier rejects cultural clutter and assigns the remaining objects (targets) to one of a finite number of categories.

Figure 2 depicts our multi-stage approach to ATD/R problem. The focus of attention block diagram in Figure 2a can be thought of as a data reduction stage because only regions of interest in the input imagery are passed to the classification stage. The focus of attention is very important in multi-stage ATD/R problems in the sense that the performance of the focus of attention stage impacts the global performance of ATD/R systems in terms of detection rates and processing powers of the systems.

This dissertation addresses the focus of attention for a multi-stage ATD/R system depicted in Figure 2b. We will consider the well known two-parameter CFAR detector [57] in a signal processing perspective. The two-parameter CFAR detector estimates the mean and variance from a locally defined region in a distance away from a test pixel and performs a thresholding function on a normalized pixel intensity difference by the local variance between the local mean and an estimated target mean under test. Here, the two-parameter CFAR test statistic can be thought of as two moment decompositions by the

local operators (two windows) by the CFAR stencil using a prior knowledge from a detection theory. A new two-parameter CFAR structure is proposed which decomposes the image based on a gamma kernel basis, which is called the γ CFAR detector. The γ CFAR statistic is a two-moment decomposition which is a projection of the image onto a set of basis functions which are the 2D extensions of the integrands of the gamma functions. These integrands are called the gamma kernels [19] [64]. There is a free parameter in this kernel set that controls the region of support of the kernels.

Then, the two-parameter CFAR detector will be further analyzed and be generalized to the Quadratic Gamma Detector (QGD) [63] which is designated to be used in the false alarm reduction stage.

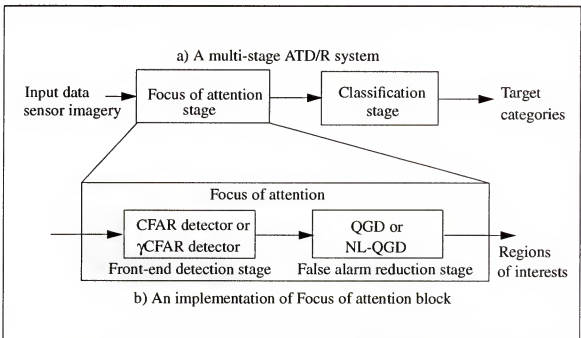


Figure 2 A multistage approach to ATD/R problems.

The construction of the QGD, inspired by the two-parameter CFAR detector, is viewed in a signal processing and pattern recognition context. The QGD effectively constructs a set of features by using a feature extractor. The feature extractor projects image intensity in local regions (and the intensity square) onto the gamma kernel basis. These

features (quadratic in the image intensity) are then classified by a linear classifier (the quadratic gamma detector QGD) or a neural network (NL-QGD).

Preliminary tests conducted at Lincoln Laboratory showed marked improvements of the CFAR/QGD with respect to the conventional CFAR detector, both with 1 foot fully polarimetric SAR data, and also with 1 meter single polarization SAR [63]. Presently this combination is used in the benchmark ATR algorithm suite at MIT/LL and also as the focus of attention in the ARPA Monitor program.

This dissertation explains the structure of the focus of attention in detail and its extensions to neural networks. We also discuss tests conducted in our laboratory with ISAR targets embedded in SAR imagery (the MIT/LL ATDS mission 90 data pass 5 data set).

1.3 Overview of the Dissertation

Chapter 2 gives a brief introduction to Synthetic Aperture Radar (SAR) and Inverse SAR (ISAR) image formation used for this study. It discusses the Polarimetric Whitening Filter (PWF) as a preprocessor for SAR data as well as a target embedding strategy.

Chapter 3 discusses a two-parameter Constant False Alarm Rate (CFAR) detector as a prescreener used for the front-end detection stage of the focus of attention stage. A gamma CFAR (γ CFAR) detector is invented as an alternative to the two-parameter CFAR detector, using a set of gamma kernel functions. In Chapter 4, the QGD is introduced for the false alarm reduction stage and is extended to the NL-QGD.

The training strategies for the NL-QGD are discussed in Chapter 5. In Chapter 6, the results of experiments measuring the performance of the focus of attention in the ATD/R system on real-life imagery (Mission 90 Pass 5 SAR data set) are discussed.

Chapter 7 concludes the study and presents a summary with recommendations for future work.

CHAPTER 2

SYNTHETIC APERTURE RADAR (SAR) DATA DESCRIPTION

2.1 Introduction

SAR is a coherent system in that it retains both phase and magnitude of the backscattered signals (echoes). SAR refers to a technique used to synthesize a very long antenna by combining echoed signals received by the radar as it moves along its flight track. The high resolution is achieved by synthesizing an extremely long antenna aperture [16]. Aperture refers to the opening used to collect the reflected energy that is used to form an image. In the case of a camera, this would be the shutter opening; for radar it is the antenna. A synthetic aperture can be therefore constructed by moving a real aperture or antenna through a series of positions along the flight track.

The net effect is that a SAR system is capable of achieving a resolution independent of a sensor altitude [16] [24]. This characteristic makes SAR an extremely valuable instrument for space observation. As an active system, SAR provides its own illumination and is not dependent on light from the sun, thus permitting continuous day/night operation, and has the additional advantage of operating successfully in all weather conditions since neither fog nor precipitation have a significant effect on microwaves, depending on the wavelengths.

There are three common SAR imaging modes: spotlight, stripmap, and scan. During a spotlight mode data collection, the sensor steers its beam antenna to continuously illuminate a terrain patch being imaged. In the stripmap mode, antenna pointing is fixed relative to the flight line, resulting in a moving antenna footprint that sweeps along a strip of terrain parallel to the pass of platform motion. In the scan mode, the sensor steers the antenna

beam to illuminate a strip of terrain at any angle to the path of motion. The scan mode is a versatile operating mode that encompasses both the spotlight and stripmap modes as special cases. Because the scan mode involves additional operation and processing complexity, spotlight and stripmap modes are the most common SAR imaging modes. The spotlight mode allows the capability of collecting fine-resolution data from localized areas. The stripmap mode is appropriate for imaging large regions with coarse resolution.

2.2 SAR Image Formation

The high resolution in radar systems can be achieved by a technique called aperture synthesis [24]. This technique enables much finer resolution to be achieved than would be possible with a conventional side-looking radar. In a side-looking radar, an antenna which is fixed parallel to the track directs a radar beam broadside and downward from the platform tracks as shown in Figure 3. The ground area that one pulse illuminates is called the radar's footprint. The beam is scanned by the motion of the platform so that the beam footprint is swept along a swath on the terrain surface. The dimension of the footprint is determined by the antenna size, the range, and the transmitted wavelength. With an antenna length L and a transmitted wavelength λ , the azimuth width of a footprint is approximately λRL at a range R [24]. The footprint of the illuminated on the ground does not disappear outside this region but fades quickly. The width λRL specifies the 3 dB level where the power of the footprint is half the maximum power. The radar receives and records the backscattered energy from the swath surface and generates an image of the surface reflectivity. The spatial (range and azimuth) resolution of the image is determined by the pulse width and the radar beam width in the range direction [24]. While the pulse width can be narrowed, and a finer range resolution achieved, the length L of the radar antenna determines the resolution in the azimuth direction of the image; the longer the antenna is the finer the resolution in this direction will be. As an example, in order to achieve a 25 meter azimuth resolution from the Seasat satellite with $\lambda = 23.5 \text{ cm}$ and $R = 850 \text{ km}$, the antennal

length requirement is $8 \text{ km} = (23.5 \text{ cm})(850 \text{ km})/(25 \text{ m})$. This is obviously a prohibitively large antenna and not practical for achieving the specified azimuth resolution. In conclusion, the goal is to synthesize an image with the resolution of a focused large aperture antenna system using the data returned from a physically small sized antenna by using SAR.

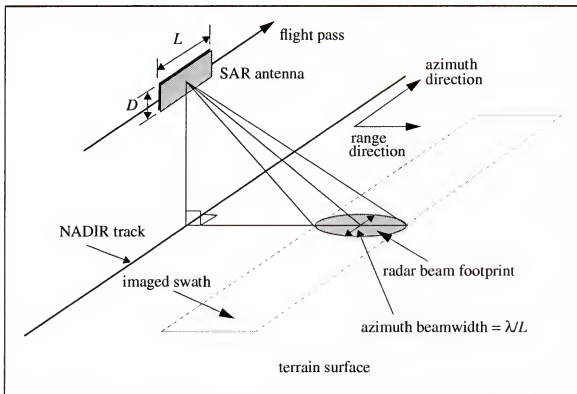


Figure 3 Imaging geometry of a side-looking aperture radar.

2.2.1 Range Processing

A real aperture can achieve the range resolution by emitting a brief intense rectangular pulse, then sampling the returned signal and averaging over time intervals no shorter than the emitted pulses. That is, the effective duration and energy of the transmitted pulse determines the range resolution and maximum range of a radar system. Shorter duration pulses allow closely spaced targets to be discriminated, while high energy pulses provide measurable reflections from targets at large ranges. In order to avoid the difficult and

expensive development of hardware to generate short duration pulses with energy characteristics, increased duration pulses are coded for transmission and then compressed at echo reception.

A linear frequency modulated (FM) waveform of a finite duration is often used for pulse coding, which is called the chirp waveform, and a correlation (matched filter) receiver is used for compression [24]. The frequency modulation enables high range resolution to be achieved at low transmitter peak power. Functions of the form $\cos\left[2\pi\left(ft + \frac{1}{2}at^2\right)\right]$, or more generally $e^{j2\pi\left(ft + \frac{1}{2}at^2\right)}$, compress into very sharp autocorrelations. For the complex exponential with phase $2\pi\left(ft + \frac{1}{2}at^2\right)$, the first time derivative of the phase of the waveform is $2\pi(f + at)$. The frequencies of the waveform changes linearly with a slope of a as time t increases. The larger the value of a the faster the frequencies change. Figure 4 depicts a chirp wave and its autocorrelation function.

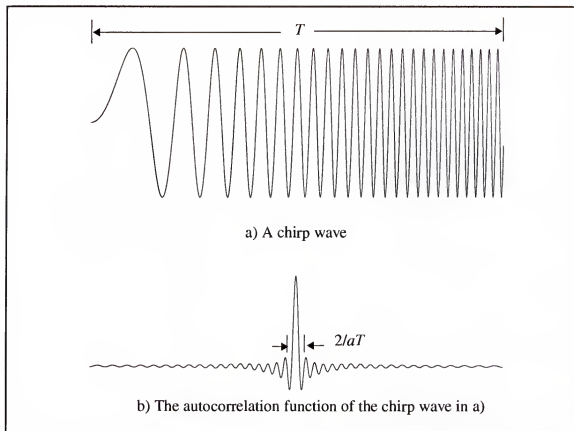


Figure 4 A chirp waveform and its autocorrelation function.

The autocorrelation $A(\tau)$ of the chirp can be calculated as

$$\begin{aligned}
 A(\tau) &= \int_{T_0}^{T_0+2T} e^{-j2\pi(ft + \frac{1}{2}at^2)} e^{-j2\pi(f(t+\tau) + \frac{1}{2}a(t+\tau)^2)} dt \\
 &= e^{j2\pi(ft + \frac{1}{2}at^2)} \int_{T_0}^{T_0+2T} e^{-j2\pi a\tau t} dt \\
 &= e^{j2\pi\tau(f + a(T_0 + \frac{1}{2}T))} \frac{\sin[\pi a\tau(T - |\tau|)]}{\pi a\tau(T - |\tau|)} \quad \text{for } -T \leq \tau \leq T \quad (1)
 \end{aligned}$$

The term $T - |\tau|$ is a triangle function weighting the $\sin(x)/x$ or sinc function. The width of the main lobe of the autocorrelation function is approximately $2/aT$ and the half power is about $1/aT$. Note that the product aT is the bandwidth of the chirp over the pulse duration T . The gain in resolution, or pulse compression ratio, is T divided by $1/aT$ or aT^2 . This is also the time-bandwidth product of the chirp signal. Thus a high time-bandwidth is required for high resolution.

For a pulse shape function, $u(t)$, and received signal of the form $r(t) = \sigma u(t - \tau)$, the receiver that implements a correlation for complex signals is given by

$$\begin{aligned}
 y(t) &= \int u^*(s) r(s + t) ds \\
 &= \sigma \int u^*(s) r(s + t - \tau) ds \\
 &= \sigma A(t - \tau) \quad (2)
 \end{aligned}$$

where σ is the target reflectivity from the range corresponding to time τ . When the pulse shape $u(t)$ is selected such that its autocorrelation fades quickly as time lag τ increases the output $y(t)$ of the receiver will be maximum when t equals τ and be small otherwise. This means that the output of the receiver will have spikes associated with time delays which correspond to reflecting objects.

In general, if there are N reflectors in a target reflecting energy, then there will be N

output spikes from the correlation receiver with each one scaled based on the reflectivity of the associated target. If $r(t)$ is given by $\sum_{i=1}^N \sigma_i u(t - \tau_i)$, then the output of the correlation receiver is

$$\begin{aligned}
 y(t) &= \int u^*(s) r(s+t) ds \\
 &= \sum_{i=1}^N \sigma_i \int u^*(s) u(s+t-\tau_i) ds \\
 &= \sum_{i=1}^N \sigma_i A(t-\tau_i)
 \end{aligned} \tag{3}$$

The output of the receiver can be expressed as the convolution of the received signal with an impulse response $h(t)$ as follow,

$$\begin{aligned}
 y(t) &= \int u^*(s) r(s+t) ds \\
 &= \int u(s) r^*(-(t-s)) ds \\
 &= \int r(s) h(t-s) ds
 \end{aligned} \tag{4}$$

The impulse response of the linear filter which implements the correlation receiver is therefore $h(t) = u^*(-t)$. This convolution implementation is descriptively called the time reversed filter which can be easily implemented in an existing linear filter architecture by modifying the impulse response and the output is then

$$y(t) = r(t) * h(t) \tag{5}$$

This filter can be implemented in the frequency domain as

$$Y(f) = R(f)H(f) = R(f)U^*(f) \tag{6}$$

where $Y(f)$, $R(f)$, and $H(f)$ are the Fourier transform of $y(t)$, $r(t)$, and $h(t)$ respectively. $U^*(f)$ is the conjugate of $U(f)$ which is the Fourier transform of $u(t)$.

The complex conjugate operator in the frequency domain corresponds to a complex conjugate together with time reversal in the time domain [59]. The filter described by the corre-

lation, time reversed, and conjugated receivers is referred to as the matched filter. This is because the filter is essentially a reference replica of the transmitted pulse which is compared to the received signal. When characteristics of the propagation medium are known, the reference waveform is given the shape of the anticipated received signal. The filter output is a measure of how precisely the received signal and reference match.

2.2.2 Azimuth processing

The range resolution in a radar system was determined by the type of pulse coding and the way in which the return from each pulse is processed. All the radar systems, conventional radars or SARs resolve targets in the range dimension in the same way. It is the resolution of targets in the azimuth dimension that distinguishes a SAR from other radar systems. The principle of SAR is to store successive echoes to a moving radar from targets in ground, and to process them to synthesis a long aperture, thereby achieving high azimuth resolution [17].

A radar with an antenna length L_a in the azimuth direction generates the radar beam that has an angular spread of $\theta_H = \lambda/L_a$ in Figure 5. Two point targets on the ground separated by an amount of δx in the azimuth direction and a slant range R can be resolved only if they are not both in the radar beam and

$$\delta x = R\theta_H = R\frac{\lambda}{L_a} \quad (7)$$

This is the resolution limit of a conventional side-looking real aperture radar. It is clear from this that the azimuth resolution capability of the conventional radar varies inversely with the physical antenna size, becoming finer for increasing antenna length L_a , and degrading with increased slant range R .

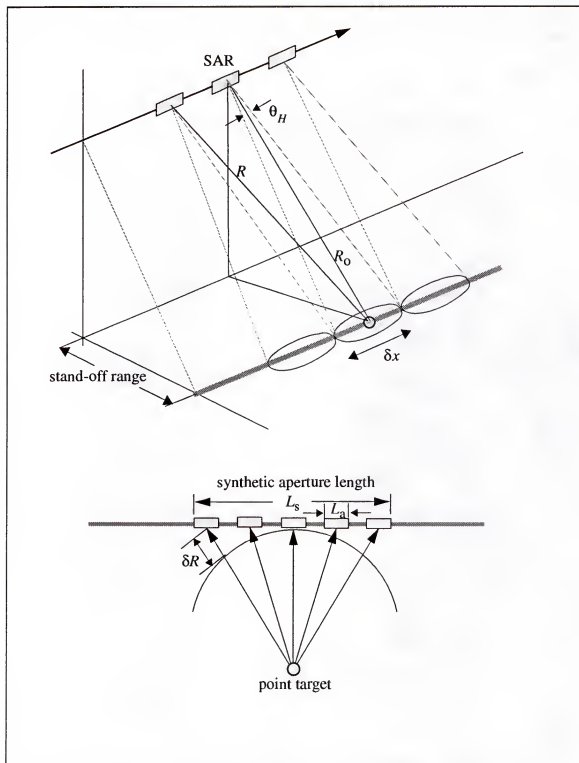


Figure 5 Aperture synthesis. The point target is in the beam for a time $T_s = L_s/V$. After phasing correcting the signals, a synthetic antenna pattern is obtained which is equivalent to that of a conventional antenna of length $2L_s$.

In SAR operation, suppose that we consider a radar beam to sweep over an arbitrary target as the platform flies over the scene in Figure 5. The point target remains in the beam for a certain time interval T_s . During this time interval the radar transmits pulses at a certain rate (the pulse repetition frequency or PRF) and also receives backscatter off of the point target during the repetition times between successive pulse transmissions. Therefore after the time interval T_s , a collection of backscatters are built up which span a spatial interval L_s , equal to the beam width.

$$L_s = \delta x = \lambda \frac{R}{L_a} \quad (8)$$

The backscatter from the point target is distributed over a large number of apertures along the track's spatial extent. A large antenna aperture has the difficulty of being physically implemented but is synthesized by sequentially gathering the backscatters using a small sized antenna at different positions which collectively define the antenna array.

The slant range R from the radar to a point target can be written

$$\begin{aligned} R &= \sqrt{R_o^2 + x^2} \\ &\approx R_o \left(1 + \frac{(vt)^2}{2R_o^2} \right) \quad \text{if } R_o \gg x \end{aligned} \quad (9)$$

where R_o is the slant range when perpendicular to the flight line and t is the elapsed time from when the platform passed its closest point of approach to the point target. Hence the phase shift of transmitted and received signals is

$$\begin{aligned} \phi &= -\frac{2\pi}{\lambda} \times 2R \\ &= \phi_o - \frac{2\pi}{\lambda} \frac{x^2}{R_o} \end{aligned} \quad (10)$$

where λ is the radar wavelength, and $\phi_o = -4\pi R_o/\lambda$. The Doppler frequency shift between the transmitted and received signals is given by

$$f_D = \frac{d\phi(t)}{dt} = -\frac{2v^2}{\lambda R_o} t \quad (11)$$

From (11), The azimuth spread of the point target response approximates a linear frequency modulated waveform. The Doppler frequency shift is highest when the point target enters the radar beam. This decreases with time until it becomes negative and reaches a minimum before the point target moves out of the area of illumination.

The azimuth resolution to resolve two consecutive point targets in the azimuth direction is determined by

$$\Delta f = \frac{2v\Delta x}{R_o\lambda} \quad (12)$$

and

$$\Delta x = \frac{R_o\lambda}{2v} \times \Delta f \quad (13)$$

The Doppler resolution of the processing is the reciprocal of the time T taken to synthesize the aperture, which is

$$T = \frac{R_o\lambda}{vL_a} \quad (14)$$

The maximum azimuth resolution is therefore the value of Δx which corresponds to a Doppler bandwidth of $1/T_s$

$$\Delta x = \frac{R_o\lambda}{2v} \times \frac{vL_a}{R_o\lambda} = \frac{L_a}{2} \quad (15)$$

(15) implies that the azimuth resolution improves as the antenna length L_a decreases. However, shorter antennas require more power for signal transmission and a longer synthetic aperture.

2.3 ISAR image formation

Besides the three SAR imaging modes mentioned earlier, there is a fourth operating mode called inverse SAR (ISAR). SAR in this mode produces radar signal data similar to

that of spotlight mode SAR. However, the ISAR mode is different in that data collection is accomplished with the radar stationary and the target moving. The signals are similar because it is the relative position and motion between the sensor and scene being imaged. Since the signals are similar, the processing required to produce an image is also similar.

Therefore, ISAR imaging refers to the use of target motion alone to generate a synthetic aperture for azimuth resolution. ISAR imaging can be accomplished by rotating the platform (turntable) on which a target is imaged. The ISAR operation is very useful for collecting the RCS (radar cross section) information of a target depending on many different aspect angles given different depression angles.

ISAR uses the same carrier term as in the SAR case but the difference is that the motion of a target relative to the radar platform is rotational as opposed to linear. The resulting Doppler phase term due to rotation can be linearized as a function of azimuth range position. Figure 6 displays an ISAR imaging scenario. The radar is stationary as the target rotates at a constant angular rotation rate ω in rad/s on a turntable. As a scatterer at a point x on the x -axis is rotated through an angle $\Delta\theta$, the change in the Doppler phase is

$$\delta f_D = \frac{2}{\lambda} \omega \delta r_c \quad (16)$$

The azimuth resolution to resolve two scatterers on the x -axis for a small viewing-angle rotation $\Delta\theta$ is obtained for $\Delta f_D = 1/T$ as

$$\begin{aligned} \Delta r_c &= \frac{\lambda}{2\omega} \Delta f_D = \frac{1}{2} \frac{\lambda}{\omega T} \\ &= \frac{1}{2} \frac{\lambda}{\Delta\theta} \end{aligned} \quad (17)$$

The range resolution for ISAR is obtained by using wideband waveforms as with SAR.

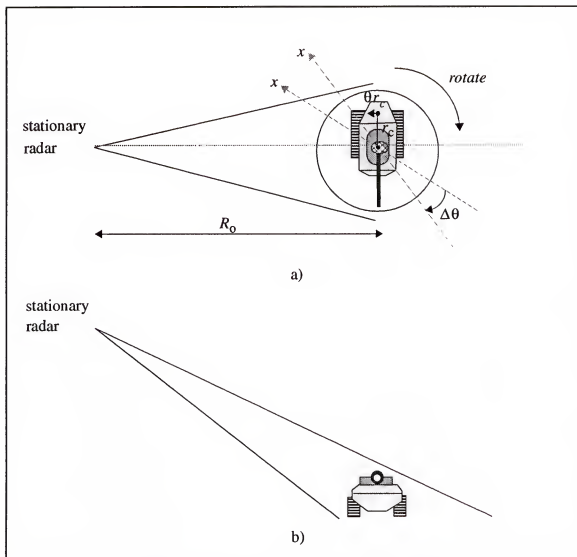


Figure 6 An ISAR imaging mode. a) top view, b) front view.

Figure 7 illustrates ISAR imaged vehicles from the TABILS 24 ISAR data set. The radar used for the data set is a fully polarimetric, K_a band radar. In Figure 7a and Figure 7b, the MV0015 and MV0095 vehicles are shown at depression angles of 20° and 15° respectively, each with aspect angles of 0° , 40° , 80° , and 120° . Target scattering looks very different, depending on the azimuth angles.

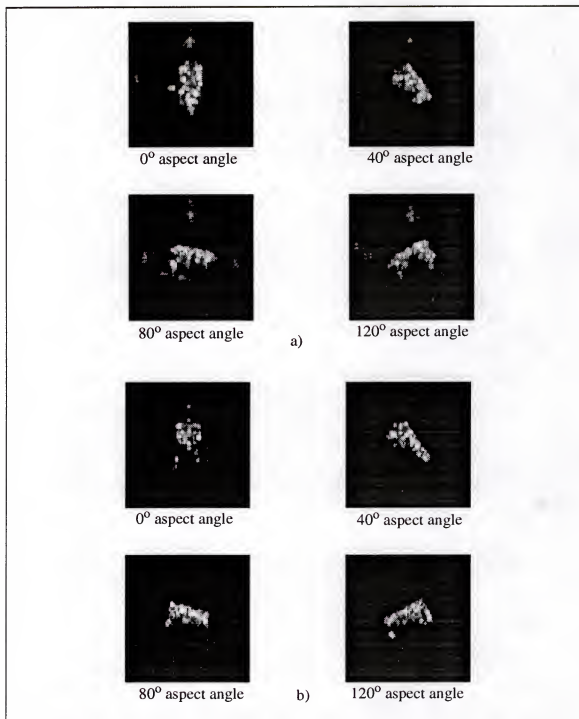


Figure 7 Examples of ISAR imagery. Down range is increasing from left to right.

2.4 Preprocessing

It is often required that before deploying detection and recognition tasks on the image, the image enhancement is often necessary to enhance target detection/recognition performance. Image enhancement is therefore viewed as a preprocessing step before proceeding ATD/R tasks.

With the availability of fully polarimetric high-resolution SAR imagery, several image enhancement techniques have been developed, exploiting the polarization scattering characteristics of targets and background clutter. SAR processing allows for high-resolution images, but introduces considerable amount of speckle in the image due to the coherent nature of the imaging process. The primary goal of preprocessing in polarimetric SAR imagery is to reduce image speckle and to improve target-to-clutter contrast.

Although many image enhancement techniques have been developed, polarimetric enhancement techniques are particularly desirable over other enhancement techniques because they can provide significant speckle reduction and target-to-clutter ratio improvement while preserving the resolution of the original SAR imagery.

Novak et al developed the polarimetric whitening filter (PWF) which combines polarimetric measurements to produce an intensity image having minimum speckle [53] [54] [55] [56]. Such an improvement led to enhance target detection performance [14] [53], clutter segmentation ability [10], and texture discrimination ability [9]. The detection algorithms discussed and developed in this dissertation are tested and evaluated based on the PWF SAR. The PWF technique developed by Novak et al. is introduced in the following sections.

2.4.1 Polarimetric Clutter Model

A mathematical model is used to characterize fully polarimetric radar returns from ground clutter. When operating in a linear polarization basis, a synthetic aperture radar uses four polarizations (HH , HV , VH and VV) to measure the full polarization scattering

matrix for any such clutter region. Since the HV has a reciprocity relationship with the VH , the set of three polarizations (HH , HV , and VV) contains all the information in the polarization scattering matrix.

A realistic assumption is often made on ground clutter and sea clutter because they are spatially nonhomogeneous. Such clutter are modeled with non-Gaussian models. The polarimetric measurement Y for each SAR pixel in ground clutter is expressed by three complex elements: HH , HV , and VV .

$$Y = \begin{bmatrix} HH \\ HV \\ VV \end{bmatrix} = \begin{bmatrix} HH_I + jHH_Q \\ HV_I + jHV_Q \\ VV_I + jVV_Q \end{bmatrix} \quad (18)$$

where HH_I and HH_Q are the in-phase and quadrature components of the complex HH measurement. Y is assumed to be the product of a complex Gaussian vector X and a spatially varying texture variable \sqrt{g} .

$$Y = \sqrt{g}X \quad (19)$$

The vector X is assumed to be zero-mean (due to the random absolute phase of its components) and complex Gaussian. Hence, the probability density function of the vector X is given by

$$f(X) = \frac{1}{\pi^3 |\Sigma|} \exp(-X^\dagger \Sigma^{-1} X) \quad (20)$$

where the symbol \dagger represents the operation of Hermitian transposition, and $\Sigma = E(XX^\dagger)$ is the polarization covariance matrix of X . In general, the polarization covariance matrix in a homogeneous region of clutter takes the form

$$\Sigma = \sigma_{HH} \begin{bmatrix} 1 & 0 & \rho\sqrt{\gamma} \\ 0 & \varepsilon & 0 \\ \rho^*\sqrt{\gamma} & 0 & \gamma \end{bmatrix} \quad (21)$$

where

$$\sigma_{HH} = E(|HH|^2) \quad (22)$$

$$\varepsilon = \frac{E(|HV|^2)}{E(|HH|^2)} \quad (23)$$

$$\gamma = \frac{E(|VV|^2)}{E(|HH|^2)} \quad (24)$$

$$\rho = \frac{E(HH \cdot VV^*)}{[E(|HH|^2) E(|VV|^2)]^{1/2}} \quad (25)$$

It is assumed that the product multiplier g is a gamma-distributed random variable. This assumption is universal: the log-normal and Weibull models are also widely used. The gamma-distributed random variable g has the form of a distribution

$$f_G(g) = \frac{1}{\bar{g}} \left(\frac{g}{\bar{g}}\right)^{(\nu-1)} \frac{1}{\Gamma(\nu)} \exp\left(-\frac{g}{\bar{g}}\right) \quad (26)$$

where the parameters \bar{g} and ν are related to the mean and the variance of the random variable g .

$$E(g) = \bar{g}\nu \quad (27)$$

$$E(g^2) = \bar{g}^2\nu(\nu+1) \quad (28)$$

where E is a statistical mean operator. Therefore, the resulting probability density function of Y is the modified Bessel function, or generalized K -distribution [52], given by

$$f(Y) = \frac{2}{\pi^3 \bar{g}^{\nu} \Gamma(\nu) |\Sigma|} \frac{K_{3-\nu} \left(2 \sqrt{\frac{Y^\dagger \Sigma^{-1} Y}{\bar{g}}} \right)}{(g Y^\dagger \Sigma^{-1} Y)^{(3-\nu)/2}} \quad (29)$$

Note that when $\bar{g} = \frac{1}{v}$ so that the mean of the texture variable is unity the non-Gaussian model, in the limit as $v \rightarrow \infty$ reduces to the Gaussian model.

The assumption that HV is uncorrelated with HH and VV is not always true (especially for man-made targets or for a polarimetric SAR with cross-talk between channels) but is valid for ground clutter [53].

2.4.2 Polarimetric Whitening Filter (PWF)

The mathematical model established in the previous section is now used for processing the polarimetric measurements HH , HV , and VV to form an enhanced SAR intensity image. An optimal processor known as the PWF is derived, which combines the polarimetric measurements to produce an intensity image having a minimum amount of speckle. A quadratic processing of the polarimetric measurement to an intensity image is constructed as follows,

$$y = Y^\dagger AY = gX^\dagger AX \quad (30)$$

where the weighting matrix A is assumed to be Hermitian symmetric and nonnegative definite, and g is a spatially varying texture variable. The objective is to find the optimal weighting matrix which leads the quadratic processing of the polarimetric measurement to produce an intensity image with a minimum amount of speckle. The ratio of the standard deviation of the image pixel intensities to the mean of the image pixel intensities is used as a measure of speckle and is given by

$$\frac{s}{m} = \frac{\text{standard deviation of } y}{\text{mean of } y} = \left[\frac{\text{VAR}(Y^\dagger AY)}{E^2(Y^\dagger AY)} \right]^{1/2} \quad (31)$$

where VAR is the variance.

Instead of minimizing the speckle amount s/m , we minimize the square of the speckle amount $(s/m)^2$.

$$\left(\frac{s}{m}\right)^2 = \frac{\text{VAR}(Y^\dagger AY)}{E^2(Y^\dagger AY)} = \frac{\text{VAR}(gX^\dagger AX)}{E^2(gX^\dagger AX)} \quad (32)$$

We use the following useful results:

$$E(X^\dagger AX) = \text{tr}(\Sigma A) = \sum_{i=1}^3 \lambda_i \quad (33)$$

$$\text{VAR}(X^\dagger AX) = \text{tr}(\Sigma A)^2 = \sum_{i=1}^3 \lambda_i^2 \quad (34)$$

where tr is the trace, and λ_1 , λ_2 , and λ_3 are the eigenvalues of the matrix ΣA . With these results, the square of the s/m ratio can be written as

$$\begin{aligned} \left(\frac{s}{m}\right)^2 &= \left[\frac{\text{VAR}(g)}{E^2(g)} \right] \times \left[\frac{E(g^2)}{E^2(g)} \cdot \frac{\text{VAR}(X^\dagger AX)}{E^2(X^\dagger AX)} + \frac{\text{VAR}(g)}{E^2(g)} \right] \\ &= \left[\frac{v+1}{v} \cdot \frac{\sum_{i=1}^3 \lambda_i^2}{\left(\sum_{i=1}^3 \lambda_i\right)^2} + \frac{1}{v} \right] \end{aligned} \quad (35)$$

Note that v is a constant in (35), and minimizing $(s/m)^2$ is equivalent to minimizing

$$\frac{\sum_{i=1}^3 \lambda_i^2}{\left(\sum_{i=1}^3 \lambda_i\right)^2} \quad (36)$$

Note from Eq 35 that if the set $\{\lambda_1, \lambda_2, \lambda_3\}$ yields a minimum for $(s/m)^2$, then so does the set $\alpha\lambda_1, \alpha\lambda_2, \alpha\lambda_3$ for any scalar α . Therefore, we can minimize (35) by minimizing its numerator $\sum_{i=1}^3 \lambda_i^2$ subject to the arbitrary constraints $\sum_{i=1}^3 \lambda_i = 3$ on its denominator. This modified optimization problem can be solved with the method of Lagrange multipliers. Using a Lagrange multiplier β , we minimize the unconstrained functional

$$f(\lambda_1, \lambda_2, \lambda_3, \beta) = \sum_{i=1}^3 \lambda_i^2 + \beta \left(9 - \left(\sum_{i=1}^3 \lambda_i \right)^2 \right) \quad (37)$$

Taking partial derivatives with respect to λ_j , and setting the results equal to zeros yields

$$\frac{\partial f}{\partial \lambda_j} = 2\lambda_j - 2\beta \sum_{i=1}^3 \lambda_i = 0 \quad \text{for } i = 1, 2, 3 \quad (38)$$

Thus we find that

$$\beta = \frac{\lambda_1}{\sum_{i=1}^3 \lambda_i} = \frac{\lambda_2}{\sum_{i=1}^3 \lambda_i} = \frac{\lambda_3}{\sum_{i=1}^3 \lambda_i} \quad (39)$$

The above result (together with the condition $\sum_{i=1}^3 \lambda_i = 3$) implies that a minimizing solution is

$$\lambda_1 = \lambda_2 = \lambda_3 = 1 \quad (40)$$

(40) leads to the following result:

$$\Sigma A = I \quad (41)$$

$$A^* = \Sigma^{-1} \quad (42)$$

(40) and (41) imply that the optimal weighting matrix A^* is the one that makes all of the eigenvalues of ΣA equal to one. The minimum speckle intensity image is therefore constructed as

$$y = gX^\dagger \Sigma^{-1} X \quad (43)$$

This solution can be interpreted as a polarimetric whitening filter. That is, the polarimetric measurement Y is transformed to a new coordinate system by the filter Σ^{-1} to obtain

$$W = \sqrt{g} \Sigma^{-\frac{1}{2}} X \quad (44)$$

This transform whitens the polarimetric measurements, so that

$$\begin{aligned} \Sigma_W &= E(WW^\dagger) = g \Sigma^{-\frac{1}{2}} E(XX^\dagger) \Sigma^{-\frac{1}{2}} \\ &= g \Sigma^{-\frac{1}{2}} \Sigma \Sigma^{-\frac{1}{2}} \\ &= gI \end{aligned} \quad (45)$$

The minimum speckle image is then obtained by noncoherently averaging the power in the elements of the whitened vector W , as shown by

$$\begin{aligned} y &= \sum_{i=1}^3 |W_i|^2 \\ &= W^\dagger W \\ &= gX^\dagger \Sigma^{-1} X \end{aligned} \quad (46)$$

In conclusion, the PWF changes the polarimetric base from a linear polarimetric base to a new base given by

$$\left[HH, \frac{HV}{\sqrt{\epsilon}}, \frac{(VV - \rho^* \sqrt{\gamma}HH)}{\sqrt{\gamma(1 - |\rho|^2)}} \right] \quad (47)$$

In this new basis, the polarimetric channels are uncorrelated and have equal expected power. Thus the optimal way to reduce speckle polarimetrically is to sum the powers non-coherently in these polarimetric channels. Another advantage of the PWF processed image is that it merges into a single image of the scatters that show up only in one of the polarization channels. Therefore, the PWF image is very useful for locating the features that might be ultimately used in a recognition task. Figure 8 summarizes the PWF processing procedure.

2.4.3 Preprocessing SAR Image

The focus of attention of the ATD/R system (Figure 2) is deployed on the PWF SAR image whose primary characteristic is that the speckle in the SAR imagery is optimally reduced [54]. Therefore, the input to preprocessing is a sequence of fully polarimetric images where each polarimetric image in the sequence is a set of three complex-valued images denoted by HH , HV and VV when the images are expressed in a linear polarization basis. Each set of three complex-valued (HH , HV , VV) pixels is optimally combined and transformed to the real-valued pixel intensity y by the PWF.

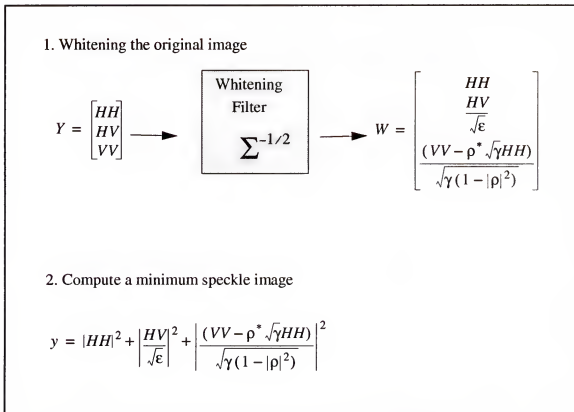


Figure 8 Minimum-speckle image processing [55].

Each set of three complex-valued (HH , HV , VV) pixels is optimally combined and transformed to the real-valued pixel intensity y by the PWF. Each pixel intensity y is related to the vector $Y = (HH, HV, VV)$ of corresponding complex polarization values HH , HV , and VV by the quadratic relation $y = Y^\dagger A Y$ where \dagger denotes the complex transposition and the matrix A is determined such that the speckle amount is minimum. Finding A requires that the polarization covariance matrix of the surrounding clutter be found.

The mission 90 pass 5 SAR imagery used in this study is a strip mode, fully polarimetric image with a linear-polarization basis. The scrub region located in the vicinity of the powerline towers in frame 105 of mission 90 pass 5 SAR data set was used for computing the covariance matrix of a typical clutter background and for the PWF processing. The estimated polarimetric covariance matrix estimated is reported to be [55]

$$\Sigma_c = 0.098 \cdot \begin{bmatrix} 1.00 + j0.00 & -0.01 + j0.02 & 0.60 - j0.05 \\ -0.01 - j0.02 & 0.19 + j0.00 & -0.00 - j0.00 \\ 0.60 + j0.05 & -0.00 + j0.00 & 1.08 + j0.00 \end{bmatrix} \quad (48)$$

2.5 SAR Image Visualization

It is often required that the image data being used as input need to be visualized on a display screen to observe the characteristics of the data such as trees, shadows, buildings. The visualization of the data is also required in the target embedding, the preparation of training and testing data for the detectors, and the selection of regions of interest.

Khoros [43] was used for such purposes, which is an integrated software development environment for information processing and visualization. Khoros worksheet brings up interactive display glyphs with pan and zoom capabilities in Figure 9. This worksheet was created for a display purpose. The worksheet loads frames and concatenates them in order. Using interactive display it allows a users to scroll through the entire plane of the loaded images and to zoom in and out on the regions of interest. Reading the current cursor position provides the corresponding coordinates (x,y) of the image.

Figure 9 displays an example of visualizing two frames of the SAR images in the Khoros working environment. One frame in the Mission 90 pass 5 SAR data set has a size of 2048 and 512 pixels in azimuth and cross range. Each pixel is a 8-bit integer value which ranges from 0 and 255. The pixel values were linearly converted from the PWF transformed data having mostly -50 dB to +30 dB.

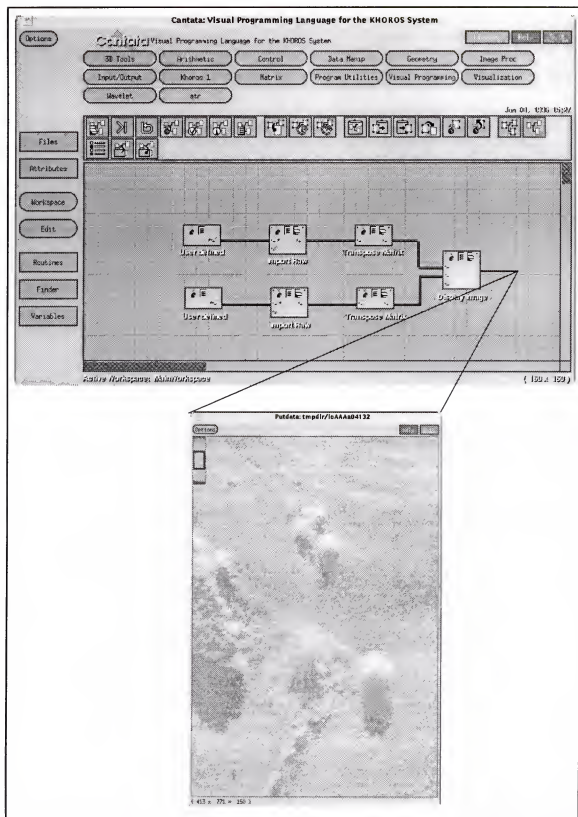


Figure 9 Khoros worksheet for a display purpose.

2.6 Examples of SAR Images

Radar images are composed of many pixel elements (pixels). Each pixel in the radar image represents the radar backscatter; brighter areas represent high backscatter. Bright features mean that a large fraction of a transmitted radar signal was reflected back to the radar receiver, while dark features imply very little reflection back from targets. Backscatter for a target area at a particular wave length depends on a variety of conditions: the size of scatters in the target area, moisture content of the target area, polarization of the pulses transmitted, the wavelength used in the radar transmitter, and observation angles.

Backscatter also depends on the use of different polarizations. Since polarimetric SARs measure the phases of incoming pulses, the phase differences (in degrees) in the return of *HH* and *VV* signals are frequently the result of structural characteristics of the scatters.

Figure 10 displays some SAR imagery preprocessed by the PWF. In general, the higher or brighter the backscatter on the image, the rougher the surface being imaged. Flat surfaces that reflect little or no radar transmitted signal back towards the radar receiver will always appear dark in radar images.

Figure 10a shows a highway and a bridge. The surfaces of the highway on the bridge appear dark since they are flat. Surfaces inclined towards the radar usually have a stronger backscatter than surfaces which slope away from the radar and tend to appear brighter in a radar image.

Figure 10b displays some houses and parked cars. The roof surfaces inclined toward the SAR appear much brighter than the surfaces inclined away from the radar. Some areas not illuminated by a radar, such as the back slope of mountains, appear dark and are called the radar shadow. As an example of this, the houses (in Figure 10b) display radar shadows at the back sides. Received radar pulses that bounced off of several objects appear very bright (white) in radar images.

Vegetation is typically rough on the scale of most radar wavelengths and appears light

grey in a radar image. Figure 10c displays scenes of powerline towers, trees and scrub. Two pairs of powerline towers are visible in the upper-left and the lower-right sides of the picture. A narrow scrub region running diagonally through the picture appears moderately coarse and the trees divided by the narrow scrub region look rough so that the trees are almost individually discernible.

Buildings which do not line up so that the radar pulses are reflected straight back will appear light grey, like very rough surfaces. Backscatter is also sensitive to the target's electrical properties, including water content. Wetter objects will appear bright, and drier targets will appear dark. The exception to this is a smooth body of water, which reflects incoming pulses away from a target; these bodies will appear dark.

2.7 Target Embedding Strategy

2.7.1 Development of a Target Embedding Method

ATR systems employ detection/recognition algorithms over the sensor imagery. A target in the imagery can exhibit an infinite number of different shapes depending on depression and aspect angles by SAR. In order to reliably test the performance of an ATR system, it is very difficult and impractical to actually place targets with many different aspect angles at many different locations over a terrain and to image the terrain.

ISAR operation provides rich shape information about a target with different view angles.

By employing an appropriate method of target embedding, targets can be placed at many different locations with many different view-angles. Figure 11 depicts a target embedding methodology. The methodology developed for target embedding proceeds as follows: (1) appropriate locations for target embedding are selected such that targets are placed in the clear and in between large scattering centers in the regions of cultural clutter, (2) in order to handle the circular-polarization-basis ISAR target images for the image

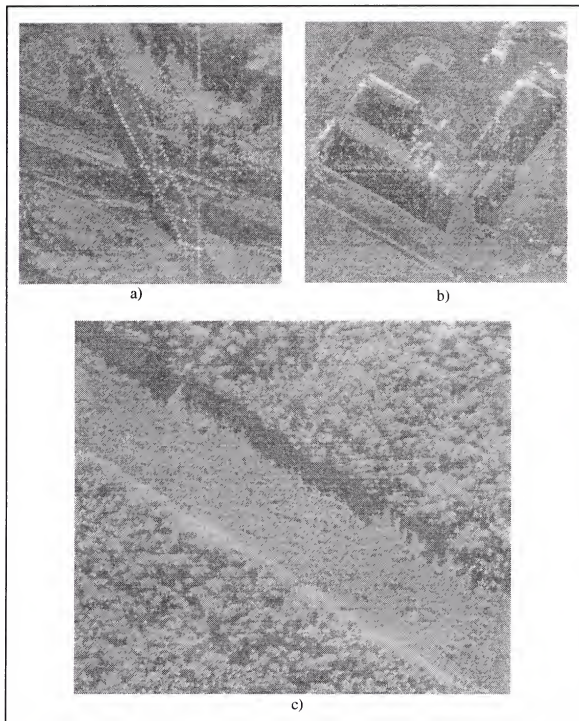


Figure 10 Some examples of SAR images from the Mission 90 pass 5 SAR data set. The images were sensed at an altitude of 2 km with a depression angle of 22.5° with a slant range of 7 km . The HH , HV and VV returns of the images were combined to produce a minimum speckle image via PWF processing. The radar sensor is located at the top of each image, looking down so that the radar shadows go downward.

(Mission 90 Pass 5 data set) in linear-polarization-basis target images, a polarization basis transformation is applied to the ISAR target images, resulting in the corresponding linear-polarization-basis target images, (3) two different images having the same polarization basis with the same resolution generated at the same radar wavelength, clutter image and target image, are *coherently added* at the locations selected for target embedding, (4) the PWF transformation is finally applied to the new image coherently added with targets.

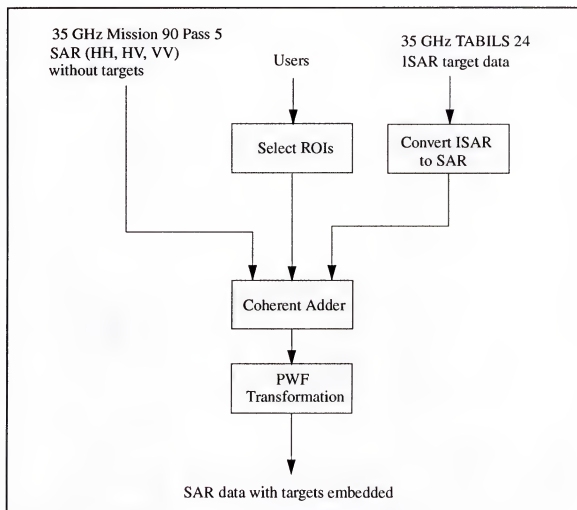


Figure 11 Target Embedding Procedure.

The *coherent addition* means that, for example, the in-phase component (HH_I) and the quadrature component (HH_Q) of a target pixel are added into the in-phase component (HH_I) and quadrature component (HH_Q) of the clutter pixel, respectively, at a location

selected for target embedding. This procedure is also applied to the other two components (HV and VV) of the pixel.

The PWF transformation leads the three complex values of each pixel to a single image pixel value. Therefore, the PWF transformed image is useful for locating the features which might be ultimately useful for recognition. After target embedding, further processing can be applied to the image with targets embedded before the detection/recognition algorithms are employed. For example, PWF processing can be applied for speckle reduction in the image.

2.7.2 Embedding the TABILS 24 ISAR Target Data into the Mission 90 Pass 5 SAR Data Set

In order to utilize and migrate plenty of targets from the TABILS 24 ISAR target data set into the strip mode SAR image, it is required that an appropriate transformation of polarization bases be applied to the ISAR target images which are circular-polarization-basis image. Therefore, the circular-polarization-basis images (i.e., *LL*, *LR* and *RR*) of the TABILS 24 ISAR targets are transformed to the corresponding linear-polarization-basis images (*HH*, *HV*, and *VV*). This transformed target images are coherently added into the appropriate locations of the mission 90 pass 5 SAR imagery selected for target embedding which is later discussed in detail.

The embedding method developed for testing performance with targets in clutter (Figure 11) combines the turntable data (TABILS 24) with the SAR data (mission 90 pass 5). No embedding method will be perfect, but it is our belief that the method we are using preserves the gross statistical characteristics of non-occluded targets in clutter upon which the pre-detection schemes in the focus of attention stage depend. Independent testing of the algorithms by the MIT Lincoln laboratory on real targets in clutter (SAR data in which targets were in the field of view during data collection) corroborated the results of the superiority of the CFAR/QGD combination [63]. So this seems to bear out the fact that the

embedding method used is sufficient for development and generation of preliminary ROCs.

We chose ISAR target data that was taken at approximately the same depression angle as the mission 90 pass 5 (23 degrees). Our assumption is that ISAR data with the same resolution of mission 90 data, and depression angles that differ from the mission 90 pass 5 depression angle by less than 3 degrees are suitable for embedding.

ISAR target images were collected from 22 target data sources of TABILS 24 data set. Since some of 22 target data sources have the same target types but were measured under different weather conditions, 22 target data sources consist of 10 different target types. For each target data source, target images were extracted at 7.5° azimuth increment over a complete of azimuth. This resulted in 1000 target images (345 training, 345 cross validation, and 345 testing). The ISAR target images within the training set, cross validation, and testing sets were separated by 7.5° azimuth increment, that is, the target images were picked up at each increment step of 7.5° and assigned to the training, cross validation, and testing sets in a sequence. The targets then need to be placed in the clear and in between large scattering centers in regions of cultural clutter.

In order to embed the targets into the clutter data, three polarizations of complex ISAR data are extracted (HH , HV , VV). These are added coherently to the different regions in the three polarizations of complex SAR data. The PWF transformation is then applied to the new data. The PWF transformed data is scaled logarithmically to the range $[0, 255]$. Since the points at which the ISAR data is embedded are range pixels with low RCS (although surrounding regions may contain range cell with very large RCS (Radar Cross Section)), the large scattering centers on the ISAR targets are not changed significantly. These are the points upon which the prescreening algorithms depend.

An example of the embedding process is shown in Figure 12. At the top of the figure is an example of an ISAR image (PWF). At the bottom left of the figure is a section of cultural clutter. At the bottom right is the same section with the ISAR target embedded. The

images are shown after PWF processing, but of course, the target is embedded prior to such processing. As can be seen the target signature is not changed dramatically after embedding. Figure 13 displays some embedded targets at a variety of aspect angles.

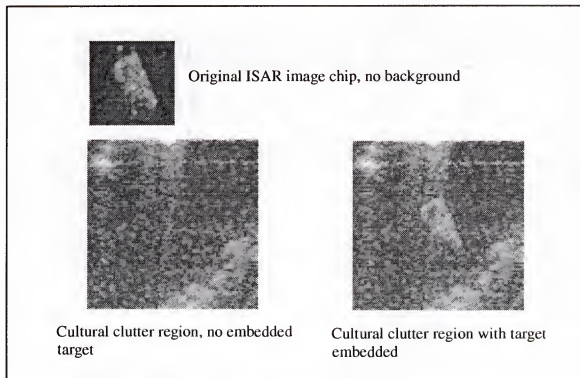


Figure 12 Target embedding.

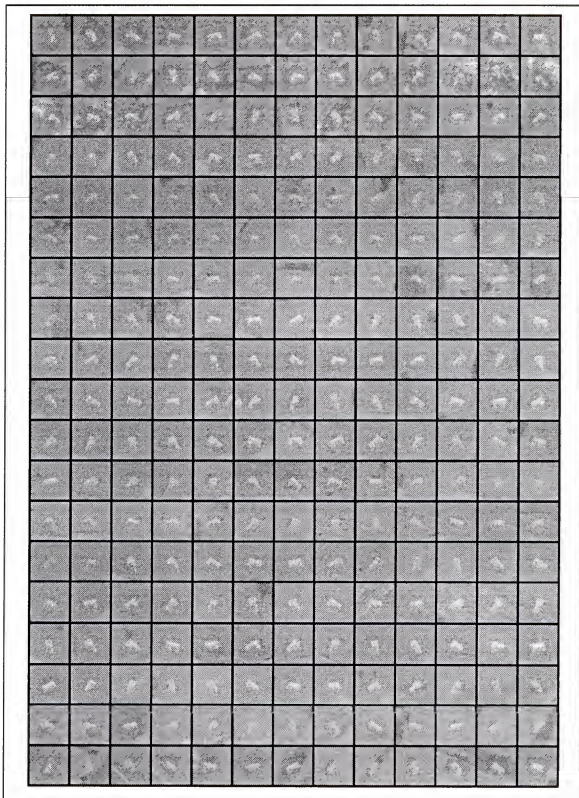


Figure 13 Some examples of the TABLIS 24 ISAR targets at variety of aspect angles after embedding.

CHAPTER 3 PRESCREENERS

3.1 Introduction

As mentioned in Chapter 1, the front-end detection stage aims to locate potential targets in the sensor imagery and allows for significant reduction of data processing in subsequent stages. Due to the direct processing of the entire imagery, the front-end detection stage requires computationally simple and efficient algorithms.

A two-parameter CFAR detector has been used as a prescreener for the front-end detection stage [57]. It computes a *Mahalanobis* distance between a pixel under test and its neighbor pixels defined in a predetermined size of window (CFAR stencil). The two-parameter CFAR detector meets the requirements of algorithm simplicity and efficiency. There is however room for improvement of the CFAR detector and we discuss this possibility of improvement. We introduce a novel detector which is called the γ CFAR detector. The γ CFAR detector extends the conventional CFAR structure by using a set of gamma kernels as an alternative to the fixed and predetermined size of the CFAR stencil.

Before we discuss the two-parameter CFAR detector, we start with a one-parameter CFAR detector to pave the road for a two-parameter CFAR detector.

3.2 A One-Parameter Constant False Alarm Rate (CFAR) Detector

A cell-averaging (CA) CFAR detector controls the detection threshold for a specific resolution cell based on the estimate of a sufficient statistic of the clutter. The detector estimates the clutter power in the cells surrounding a cell under test. When the clutter is statis-

tically homogeneous over the resolution cells, this detector works well, otherwise, the performance degrades.

In order to implement a CA operation, a stencil can be used locally for a particular location in the image (Figure 14). A test pixel is defined at the center area (R_t) of the stencil and the clutter intensity mean is estimated in a local region (R_c) in a distance away from the pixel under test. We refer to the stencil as the CFAR stencil. The output of the CA-CFAR operation can be expressed as

$$y = \frac{1}{N_t} \sum_{i,j \in R_t} x(i,j) - \frac{1}{N_c} \sum_{i,j \in R_c} x(i,j) \quad (49)$$

where x represents the intensity pixels and N_t and N_c are the numbers of pixels in R_t and R_c in the stencil. This CA-CFAR detector can be called a one-parameter CFAR detector because only the mean information is utilized in the operation in (49).

The hypothesis testing is then performed over the outputs after the CA-CFAR processing such that

$$\begin{array}{ccc} & \textit{target} & \\ y & \geq & T_{CFAR} \\ & \textit{non-target} & \end{array} \quad (50)$$

From a signal processing perspective, the one-parameter CFAR detector in (49) is a band pass filter running over the image in which the outputs of a low pass filter by the target mask of the template is subtracted from the outputs of another low pass filter by the clutter mask. Abrupt changes in image intensities produce large outputs by the one-parameter CFAR detector. Since man-made objects are usually strongly scattered back to millimeter SARs, the large intensity differences between pixels under test and their clutter background are easily detected.

However, an output of the one-parameter CFAR processing in (49) is the result of convolving the image with two rectangular-shaped kernels which cause large losses or ripples outside their main lobes in the frequency domain. Contrast enhancement between tar-

gets and clutter can be achieved by incorporating smooth-shaped kernels in the stencil [26].

The threshold of the one-parameter CFAR detector in (49) is sensitive to the scale of the image in which the threshold is linearly proportional to the scale factor of the image. In the next section, a two-parameter CFAR detector is introduced which incorporates a normalization factor. This factor is a standard deviation of the clutter in R_c .

3.3 A Two-Parameter CFAR Detector

A two-parameter CFAR detector was first developed for use on 1-D range profile by Goldstein [29]. Later, Novak et al. extended it for use in SAR imagery [56] [57].

The two-parameter CFAR detector has been commonly used as a pre-screener in multi-stage SAR ATD/R systems. Its popularity is due to an excellent figure of merit in terms of performance/simplicity. The name indicates that a constant false alarm probability of detection is achieved, but in fact this is only true for Gaussian distributed targets and clutter [57]. Normally in SAR imagery this is not the case for targets in clutter. Nevertheless, experience has shown that the two parameter CFAR is a robust detector for man-made clutter and targets [57].

The reasons for this success can be found in the simplicity and the discriminating power of the test. Basically, the CFAR compares the intensity of a pixel under test with the normalized intensity of a surrounding area. Since man-made objects are normally bright in SAR imagery, this is a very effective test, which can be efficiently implemented in digital hardware. In terms of statistical detection theory, the CFAR is estimating the parameters of the local intensity probability density function (pdf), and making a decision when there is a brightness deviation of the pixel under test with respect to the normalized mean background intensity, i.e.

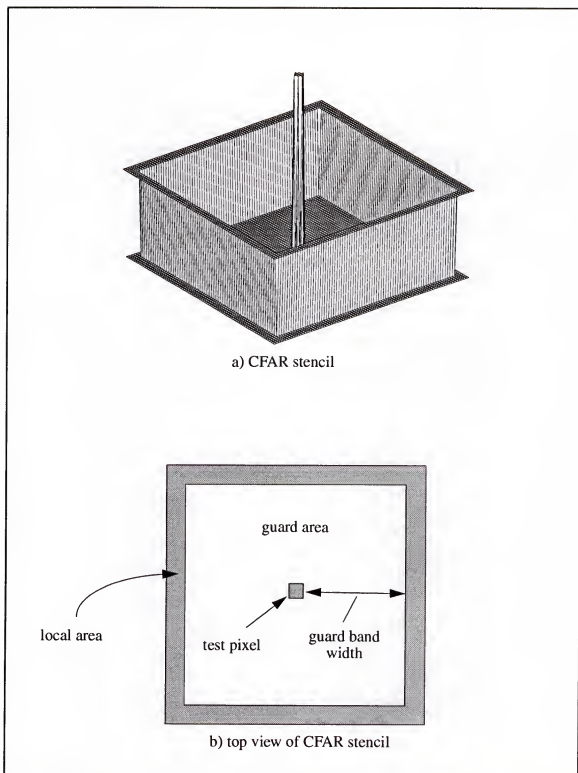


Figure 14 CFAR stencil. The amplitude of the test pixel is compared with the mean and the standard deviation of the local area. The guard area ensures that no target pixels are included in the measurement of the local statistics [57].

$$\frac{X_t - \bar{X}_c}{\hat{\sigma}_c} \underset{\text{clutter}}{\overset{\text{target}}{\geq}} T_{CFAR} \quad (51)$$

where X_t is a pixel under test, T_{CFAR} is a threshold for the two-parameter CFAR detector and \bar{X}_c and $\hat{\sigma}_c$ are the estimates of the local statistics of mean and standard deviation measured in a defined local area by the CFAR stencil (see Figure 14). The estimates, \bar{X}_c and $\hat{\sigma}_c$ are computed by

$$\bar{X}_c = \frac{1}{N_c} \sum_{c,i,j \in \Omega_c} x(i,j) \quad (52)$$

where $x(i,j)$ is the pixel intensities at (i,j) locations and

$$\hat{\sigma}_c = \sqrt{\frac{1}{N_c} \sum_{c,i,j \in \Omega_c} (x(i,j) - \bar{X}_c)^2} \quad (53)$$

where Ω_c defines the local area where \bar{X}_c and $\hat{\sigma}_c$ are computed, and N_c is the number of pixels in Ω_c .

The shape of the stencil ensures that when the center pixel is on target, the neighborhood falls in the background such that its local statistics can be reasonably well estimated. The shape of the stencil (in particular the guardband) is governed by the target size [54]. In SAR imagery, the reflectivity of the object is only weakly coupled to its geometric shape, so a priori stencil dimensions based solely on target size cause suboptimal performance.

In terms of statistical pattern recognition, we can interpret the CFAR in a slightly different way. One can think that the CFAR stencil is extracting *intensity features* in the neighborhood of the pixel under test. In fact, the CFAR equation can be rewritten as

$$X_t^2 - 2X_t\bar{X}_c + \bar{X}_c^2 - T_{CFAR}^2\bar{X}_c^2 + T_{CFAR}^2\bar{X}_c^2 \geq 0 \quad (54)$$

where \bar{X}_c^2 and $\bar{X}_c^2 (= \frac{1}{N_c} \sum_{c,i,j \in \Omega_c} x^2(i,j))$ are the square of an estimated mean and an esti-

mated mean of intensity power respectively, both of which are measured in a neighboring area selected by the CFAR stencil.

Notice that this expression is a linear combination (with fixed weighting) of the image intensity and its square at the pixel and the mean, power, and mean square of the intensity at the neighborhood. Hence we can interpret the CFAR as implementing a “restricted” linear discriminant function of quadratic terms of the image intensity.

From this perspective, the two parameter CFAR can be improved: first, it uses only some of the quadratic terms of the intensity on the pixel and its surroundings; second, it implements a fixed parametric combination of these features; and thirdly, there is little flexibility in the feature extraction because the kernel is ad-hoc. These three aspects can be greatly improved if more mathematically oriented features are computed and if trainable classifiers are built. In Chapter 4, this is exactly what the quadratic gamma detector (QGD) and even more the NL-QGD (Nonlinear Extension to QGD) will provide.

3.4 Extension to the Two-Parameter CFAR Detector

3.4.1 Introduction

In this section, the conventional CFAR detector discussed in the previous section is improved, incorporating a new stencil which is called the γ CFAR (gamma CFAR) stencil. In Section 3.4.2, the gamma filter and gamma kernel [19] [64] are reviewed. They constitute the basis of the new stencil. The 2-D extension of 1-D gamma kernels are introduced in Section 3.4.3.

As extensions to the conventional CFAR detector, γ CFAR detectors are introduced and their characteristics are discussed in Section 3.4.4.

3.4.2 Gamma Filter and Gamma Kernel

The gamma kernel was originally developed for time series analysis as a short-term memory network structure for sequence processing neural network [19]. With the property

of completeness in Hilbert space (L^2 space), any finite energy signals can be approximated arbitrarily closely using a finite number of gamma kernels. The gamma kernels are defined as

$$g_{k,\mu}(t) = \frac{\mu^k}{(k-1)!} t^{k-1} e^{-\mu t} \quad (55)$$

The implementation of the gamma kernels is accomplished by introducing a local feedback loop around the delay operator in Figure 15. The impulse response from the input to the k th tap generates the k th order gamma kernel by (55). This structure is called the gamma filter. The Laplace transform of the gamma kernels is given as

$$G_k(s) = \left(\frac{\mu}{s+\mu} \right)^k = G^k(s) \quad (56)$$

where

$$G(s) = \frac{\mu}{s+\mu} \quad (57)$$

The k th order gamma filter is characterized by a pole at $s = -\mu$ with multiplicity k . The location of zeros are determined by the weights w_1, w_2, \dots, w_k and the parameter μ . The discrete time gamma filter is depicted in Figure 15b. The discrete time gamma kernels and their corresponding z -transform are given by

$$g_k(n) = \binom{n-1}{k-1} \mu^k (1-\mu)^{n-k} \quad \text{for } n \geq k \quad (58)$$

$$G_k(z) = \left[\frac{\mu}{z - (1-\mu)} \right]^k = G^k(z) \quad (59)$$

The gamma filters in both continuous and discrete time are locally recurrent, globally feedforward structures.

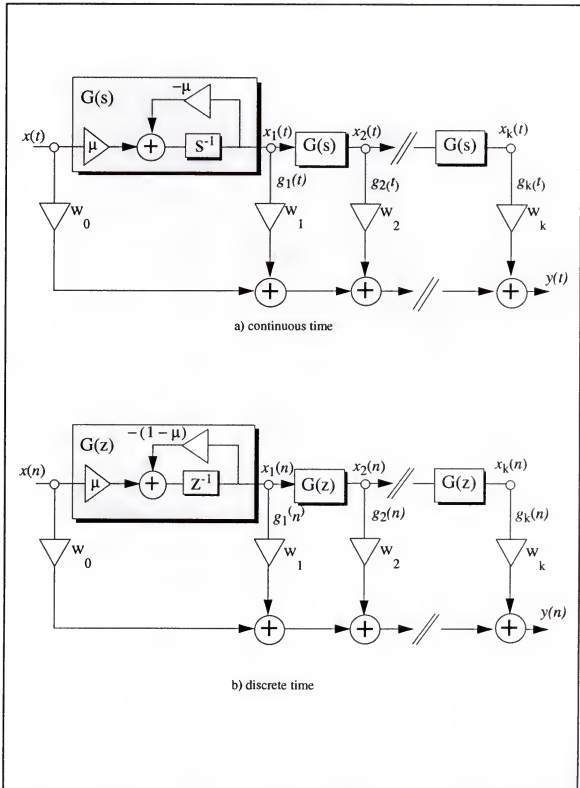


Figure 15 The gamma filter structures a) in continuous time and b) in discrete time.

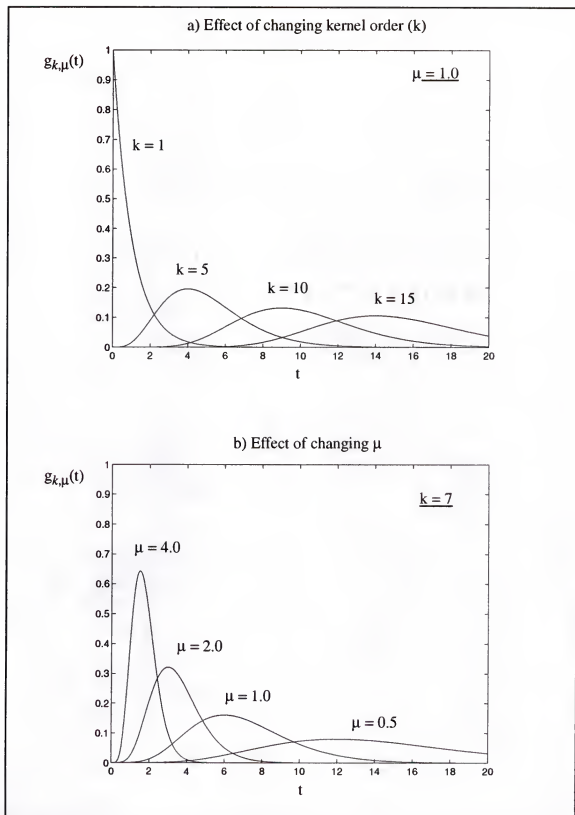


Figure 16 Shape of gamma kernels effected by parameter and kernel order (μ and k).

The discrete gamma filter generalizes a tapped delay line structure. For the case of $\mu = 1$, the gamma filter reduces to a transversal filter (or a finite impulse response (FIR) filter). The stability of the continuous time gamma filter is guaranteed because the system poles always exist in the left half plane of Laplace domain as long as $\mu > 0$ is guaranteed. For the discrete case, the poles are located inside the unit circle in Z-domain as long as $0 < \mu < 2$ is guaranteed. So the stability problem in the discrete time is also trivial. The interesting point is that the functions constitute a non-orthogonal basis that is complete for signals of finite energy (in L^2) [19]. Hence we can interpret the gamma filter output as a projection of the input signal in a linear space defined by the convolution of the input with the basis functions [20].

Figure 16 illustrates, in continuous time, an example of gamma kernel shapes as functions of the kernel order k and parameter μ . Note that the shape of the kernels is very similar when one selects the order for a fixed μ , or when μ is selected for a given kernel order.

The main characteristic of the gamma filter is that time in the filter taps is scaled by the feedback parameter μ (linear time warping). In other words, the region of support of the impulse response is controlled by the single parameter μ such that by changing μ the impulse response can be stretched out or shrunk as a rubber band (Figure 16b). In a signal processing framework, the parameter μ can be adapted with the output mean square error using a gradient descent approach, so that the best local features are captured by the filter [64].

The linear filters are often characterized in the frequency domain in terms of their magnitudes and phase responses. For the temporal processing of signals, linear filters such as the FIR, IIR and gamma filter can be understood as memory filters which are characterized by the memory depth and memory resolution of the filters. The memory depth D is defined as the temporal mean value of the impulse response $g(t)$ of a filter used for storing temporal information [19].

$$D = \int_0^{\infty} t g(t) dt = -\frac{d}{ds} G(s) \Big|_{s=0} \quad \text{for continuous time} \quad (60)$$

$$D = \sum_{n=0}^{\infty} n g(n) = -z \frac{d}{dz} G(z) \Big|_{z=1} \quad \text{for discrete time} \quad (61)$$

The memory resolution R is defined as the number of degrees of freedom (i.e. memory state variables) per unit time. As a memory filter, the impulse response of a k th order FIR filter at each tap is $g_k(n) = \delta(n-k)$. The transfer function in the Z -domain is given by $G(z) = z^{-k}$. The memory depth D of the filter is $D = K$. That is, the memory depth of an FIR filter is equal to the filter order. Due to the limitation of memory depth by the filter, a low order FIR filter has a poor modeling capability of low-pass frequency bounded signals [64]. On the other hand, the structure of an IIR filter contains feedback connections, and consequently the memory depth is not limited by the filter order. The IIR filters, however, suffer from stability problems.

The memory depth of a k th order gamma filter for both discrete and continuous time domains is obtained by (61)

$$D = \frac{k}{\mu} \quad (62)$$

Contrary to an FIR filter, a gamma filter has a memory depth uncoupled to the filter order. The gamma filter shares the property of IIR filters in memory depth due to locally introduced feedback between taps. This also relaxes the stability problem as long as the local feedback parameter is limited in the range of $0 < \mu < 2$ for discrete case.

The memory resolution of a k th order discrete time gamma filter is equivalent to the number of taps divided by the memory depth.

$$R = \frac{k}{D} = \mu \quad (63)$$

(63) can be written as

$$k = D \times R \quad (64)$$

For a given order k , the trade-off between memory depth and memory resolution is possible so that a very deep memory structure can be obtained at the expense of a low memory resolution. In an FIR case, the memory depth is the filter order and this is a special case of $\mu = 1$ in the gamma filter.

3.4.3 2-D Extension of 1-D Gamma Kernels

Theoretically the gamma filter can be extended to N dimensions without problems. This is done by simply substituting t in (55) with an N dimensional basis vector. In this extension, circularly symmetric gamma kernels are obtained but more general cases can be considered. Specifically for our applications to 2-D (image) data, the gamma kernels can be extended by

$$g_{n,\mu}(k, l) = C g_{n,\mu}(t) \Big|_{t=\sqrt{k^2+l^2}} \quad (65)$$

where the constant C is a normalization factor. The resulting 2-D kernel has a circularly symmetric shape given by

$$g_{n,\mu}(k, l) = \frac{\mu^{n+1}}{2\pi n!} (\sqrt{k^2+l^2})^{n-1} e^{-\mu\sqrt{k^2+l^2}} \quad (66)$$

$$\Omega = \{ (k, l) ; -N \leq k, l \leq N \}$$

where Ω is the region of support of the kernel, k the kernel order, and μ the parameter that controls the shape and scale of the kernel.

As a result the resulting 2-D gamma kernels are circularly symmetric. Since the 2-D circularly symmetric gamma kernels are created from the corresponding 1-D gamma kernels in the spatial domain, they preserve the spatial characteristics of the 1-D gamma kernels. That is, the concept of a time warping parameter extrapolates to the spatial domain as a scale parameter that controls the region of support of the 2-D gamma kernels.

If the 2-D gamma kernel (in (66)) defined in the Cartesian coordinate system is converted into the circular coordinate system, the memory depth of the 2-D gamma kernel can

be easily obtained. By letting $k = r \cos \theta$, $l = r \sin \theta$ and considering $dk = r dr$, $dl = d\theta$, (66) in the circular coordinated system is given by

$$g_{n,\mu}(r, \theta) = \frac{\mu^{n+1}}{2\pi n!} r^n e^{-\mu r} \quad (67)$$

The memory depth in 2-D spatial domain is expressed as

$$D = \int_0^{2\pi\infty} \int_0^{\infty} r g_{n,\mu}(r, \theta) dr d\theta = \frac{n}{\mu} \quad (68)$$

(68) is equivalent to the 1-D case. This means that the spatial characteristics of 1-D gamma kernels is preserved in a 2-D domain by the circularly symmetric rotation.

Figure 17 shows the characteristics of 2-D gamma kernels in the spatial domain. The 1st order ($n = 1$) kernel has its peak at the pivot point (0,0) with an exponentially decaying amplitude. All the other kernels have a peak at the radius n/μ , creating concentric smooth rings around the pivot point (Figure 17). For a fixed order ($n = 15$) the radial distance where the kernel peaks is still dependent upon the parameter μ , as in the 1-D case.

3.4.4 Gamma CFAR (γ CFAR) Detector

By analogy to the CFAR stencil, any combination of the first kernel with one of the higher order kernels produces a similar stencil, although the shapes with the 2-D gamma kernels are smoother. We call this stencil the γ CFAR.

It is interesting to contrast the γ CFAR detector with the CFAR detector. Target masking in the CFAR stencil is implemented by the first order 2-D gamma kernel so that target intensity mean is estimated closely around the center pixel under test, while a higher order kernel creates a clutter masking so that the statistics of clutter are measured in the round-shaped ring. With the γ CFAR, we have a better handle on the shape of the kernel due to its analytic formalization. Figure 18 shows the γ CFAR stencil for CFAR test.

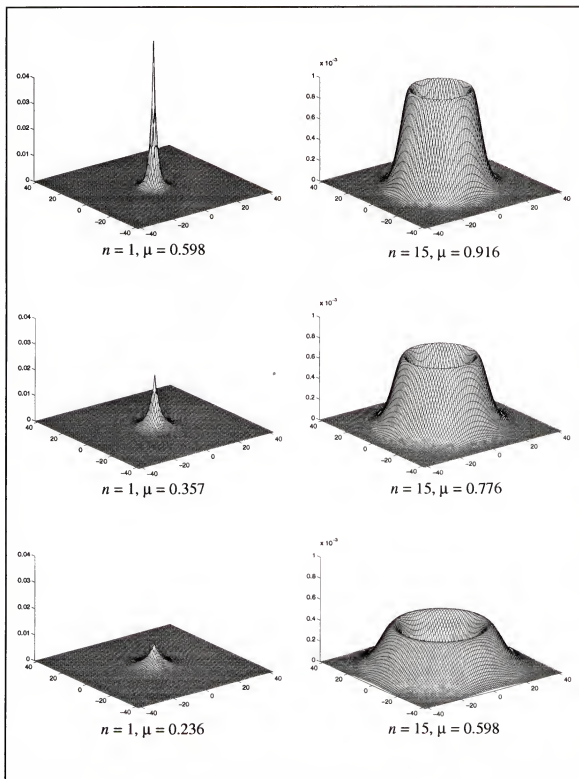


Figure 17 2-D gamma kernels.

Since the parameter μ in 2-D has exactly the same function as the 1-D counterpart, i.e., it shrinks or stretches the region of support of the image response, we can adaptively select μ to better perform CFAR test.

In fact, after fixing the order of the kernel, we have a single parameter that controls its spatial extent, and we can derive equations that will change the parameter μ to minimize an output error. Figure 19 shows the block diagram of a one-parameter γ CFAR detector. Two gamma kernels are linearly combined to form an output y of the one-parameter CFAR detector. We call this the one-parameter γ CFAR detector a counterpart for the one-parameter CFAR detector in (49). The output can be written as

$$y = w_1 g_{m, \mu_m} \bullet X + w_2 g_{n, \mu_n} \bullet X \quad (69)$$

where \bullet represents a convolution operator, w_1 and w_2 are weights, the kernel order $m = 1$ and $n > 1$, μ_m and μ_n are the parameters that control the extent of the kernel. It is expected that w_2 needs to be negative so that the output will be high over areas of largest contrast as is the case in the one-parameter CFAR detector.

In addition to the degrees of freedom associated with the controllability of μ for the kernel extent, the smooth shape of the gamma kernels yields smaller sidelobes than those associated with the CFAR stencil. From a signal processing perspective, as is the case in the one-parameter CFAR detector, the one-parameter γ CFAR detector performs bandpass filtering but has an ability of correctly choosing the frequency bands of the bright pixel intensities in the image by adapting the region of support of the kernels, and the energy of pixel intensities can be better preserved in the selected frequency band with less loss due to the smoothness of gamma kernels.

Interpreted from the projection point of view, the input signal (an image in the 2-D case) is still being projected onto a local basis obtained by convolving the input with the first kernel ($m = 1$) and a higher order kernel ($n > 1$).

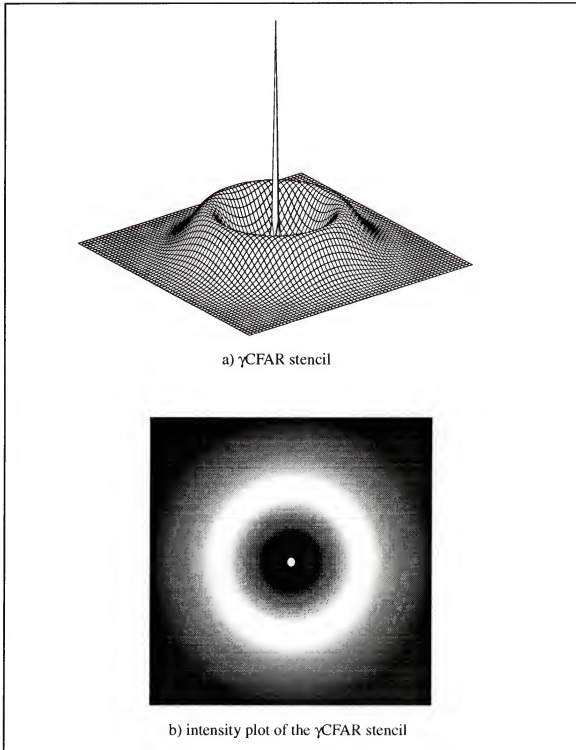


Figure 18 The γ CFAR detector: a) the center kernel has an order of 1 and the rounding kernel is of an order 15; b) the rounding kernel defines a local area where the local statistics of mean and standard deviation are measured. The peaky kernel averages a pixel under test and the very closely neighbored pixels around a pixel under test.

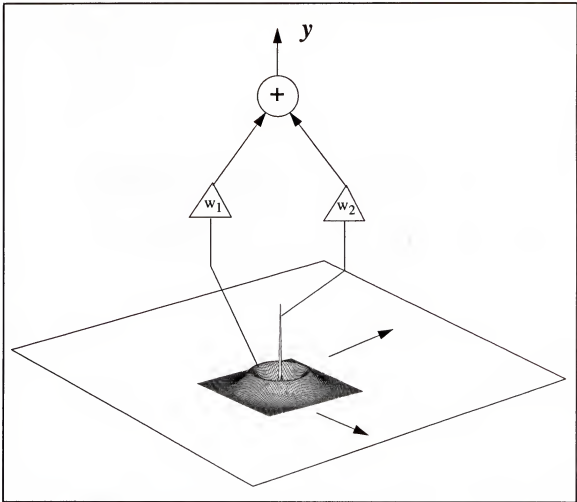


Figure 19 One-parameter γ CFAR detector.

Due to the exponential amplitude decay from the pivot point, the spatial extent of the kernel can be truncated to a small value. This projection on gamma kernels of order m and n will not be complete, meaning that the input image can not be recovered from the projection. But as the CFAR experience shows, it will still preserve the important features to distinguish man-made objects from clutter.

The decision rule in the γ CFAR detector is defined as

$$\frac{g_{m, \mu_m} \bullet X - g_{n, \mu_n} \bullet X}{\hat{\sigma}} \begin{matrix} target \\ \succ \\ clutter \end{matrix} T_{\gamma CFAR} \quad (70)$$

where \bullet represents the inner product operator, $T_{\gamma CFAR}$ is a threshold for determining target or clutter, and

$$\tilde{\sigma} = \sqrt{g_{n,\mu_n} \bullet X^2 - (g_n \bullet X)^2} \quad (71)$$

As shown in (70) and (71), the local mean and standard deviation are computed by a higher order kernel ($n > 1$) and the test pixel under test is averaged with its adjacent pixels less emphasized away from the pivot point. The $\gamma CFAR$ decision can also be recast as a discriminating function

$$\begin{aligned} (g_m \bullet X)^2 - 2(g_m \bullet X)(g_n \bullet X) + (g_n \bullet X)^2 \\ - T_{\gamma CFAR}^2 (g_n \bullet X^2) + T_{\gamma CFAR}^2 (g_n \bullet X)^2 \geq 0 \end{aligned} \quad (72)$$

It is instructive to analyze experimentally the dependency of the accuracy of the CFAR test as a function of the size of the stencil, now that we have a parametric form for the stencil.

The local image features computed by the convolution can be specified as projections of intensity and power onto gamma kernels,

$$g_{n,\mu} \bullet X = \sum_{i,j \in \Omega} g_{n,\mu}(i,j) x(i,j) \quad (73)$$

$$g_{n,\mu} \bullet X^2 = \sum_{i,j \in \Omega} g_{n,\mu}(i,j) x^2(i,j) \quad (74)$$

Note that these operations can be interpreted as FIR filtering with gamma stencil $g_{n,\mu}(i,j)$. The output values can be viewed as estimates of local 1st and 2nd moments respectively, which are sufficient to compute the local variance. The local standard deviation in (53) is needed to compute the two-parameter CFAR statistics, and has been shown to be an important measure in deciding whether a target is present or not [57].

The $\gamma CFAR$ detector separates targets from clutter based on the values of 4 features

$(g_m \bullet X)^2$, $(g_m \bullet X)(g_n \bullet X)$, $(g_n \bullet X)^2$, and $g_m \bullet X^2$ which are needed to compute local mean and standard deviation. These terms enable a one-to-one comparison of the γ CFAR detector and the two-parameter CFAR detector, and are also used to study the benefits of using adaptive feature extraction and adaptive weights in Chapter 4.

In conclusion, the potential advantage of the γ CFAR detector implementation is that the extent of the localized mean and surrounding areas can be adaptively set. This could lead to fine tuning the area where local statistics are measured. The γ CFAR detector can be a promising device based on 2-D gamma filters as an extension of the conventional CFAR detector.

3.4.5 Implementation of the γ CFAR Detector

Computing the feature set at each pixel of the image amounts to correlating each of the two kernels $g_{m,\mu}$ and $g_{n,\mu}$ with both the original image and the image squared. Each kernel assumes a role of an FIR filter with rectangular support (size $(N+1) \times (N+1)$). The three base features $(g_{m,\mu} \bullet X)$, $(g_{n,\mu} \bullet X)$, and $(g_{n,\mu} \bullet X^2)$ at a point (i, j) in the image are then obtained using a translated gamma kernels as

$$(g_{n,\mu} \bullet X)_{i,j} = \sum_k \sum_l g_{n,\mu}(k+i, l+j) x^p(i, j) \quad p = 1 \text{ or } 2 \quad (75)$$

$$\Omega = \{ (k, l); -\frac{N}{2} \leq k, l \leq \frac{N}{2} \}$$

where n stands for the kernel order and p indicates either the first or the second moment. Correlations are computed in the frequency domain using FFTs to obtain better computational efficiency. For processing large images and to avoid memory problems we divide the image into overlapping radix 2 windows which are individually processed and combined using an overlap and save method [59].

3.5 Receiver Operating Characteristic (ROC)

A plot of detection probability P_d as a function of false alarm probability P_f is referred to as receiver operating characteristic (ROC) curve. The ROC is a very important assessment for detection systems and depends on the probability density function of a observed signal under each hypothesis H_j , that is, $p_{Y|H_j}(y|H_j)$, $j = 0, 1$. Figure 20b depicts an ROC curve with the probability functions for two hypotheses. In Figure 20a, varying a threshold κ controls the area representing P_d and P_f . The corresponding ROC curve is shown in Figure 20b.

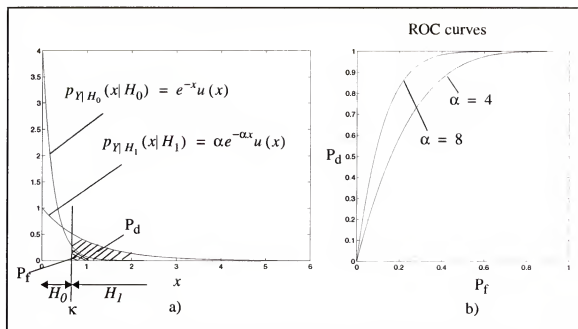


Figure 20 ROC: a) probability density functions, b) ROC curves.

CHAPTER 4

QUADRATIC GAMMA DETECTOR

4.1 Introduction

The goal of a prescreener in the front-end detection stage was to locate potential targets and to eliminate a large amount of clutter in the sensor imagery. The output of the prescreener are regions of interest which need be further considered in the following stage, false alarm reduction stage. The regions of interest are indicated by the coordinates (x,y) in the image and the further processing is applied to the locations reported by the prescreener.

Section 4.2 briefly reviews the discriminant functions. Section 4.3 develops a novel detector (QGD) [63], which is extended from the two-parameter CFAR and γ CFAR detectors, and discusses its discrimination power.

In Section 4.5 a multilayer perceptron (MLP) structure is reviewed and its training and the generalization ability of the network is also discussed. Finally the QGD is extended into a nonlinear structure such MLPs, which is called the NL-QGD (Nonlinear extension to the QGD) in Section 4.6.

4.2 Discriminant Functions

4.2.1 Linear Discriminant Functions

In general, a discriminant function is an operator that, when applied to a pattern, yields decisions concerning the class membership of the pattern. The action of a discriminant function is to produce a mapping from pattern space to attribute space.

A general linear discriminant function in n -dimensional pattern space is of the form

$$\begin{aligned} g(X) &= w_1x_1 + w_2x_2 + \dots + w_nx_n + w_{n+1} \\ &= W^T X + w_{n+1} \end{aligned} \quad (76)$$

where W is called the weight vector and w_{n+1} the threshold weight. The decision rule of a two-category classifier is implemented by the following property

$$\begin{aligned} g(X) &= W^T X + w_{n+1} \\ &> 0 \quad \text{if } X \in \omega_1 \\ &\leq 0 \quad \text{if } X \in \omega_2 \end{aligned} \quad (77)$$

The input pattern vector X is categorized into the class ω_1 if $g(X) > 0$ and into ω_2 if $g(X) < 0$. For a multicategory case, linear discriminant functions implement the decision rule: assign X to ω_i if $g_i(X) > g_j(X)$ for all $j \neq i$ where $g_i(X) = W^T X + w_{n+1}$. In case of ties, decision is left undefined.

4.2.2 Generalized Linear Discriminant Functions

The linear discriminant function discussed in the previous section can be extended to the generalized linear discriminant function of the form

$$\begin{aligned} g(X) &= w_1f_1(X) + w_2f_2(X) + \dots + w_nf_n(X) + w_{n+1} \\ &= W^T F \end{aligned} \quad (78)$$

where $F = [f_1(X) \ f_2(X) \ \dots \ f_n(x) \ 1]^T$ are a real valued vector with function elements of the pattern X , and W is a n -dimensional weight vector. Any desired discriminant function can be approximated by a series expansion $\{f_i(X)\}$ by selecting these functions judiciously and letting n be sufficiently large.

The discriminant function in (78) is not linear in the original input pattern space, but linear in the transformed pattern (F) space. The pattern F is therefore separated in the transformed space by a hyperplane but the pattern X by the hypersurface in the original pattern space. Therefore, the mapping from X to F reduces the problem to one of finding a

linear discriminant function.

4.3 Extension to the γ CFAR Detector

Quadratic discriminant functions implement the optimal classifier for Gaussian distributed classes [8] [27]. A quadratic discriminant function in d dimensional space is

$$g(X) = \sum_{j=1}^d w_{jj}x_j^2 + \sum_{j=1}^{d-1} \sum_{k=j+1}^d w_{jk}x_jx_k + \sum_{j=1}^d w_jx_j + w_{d+1} \quad (79)$$

where w is a set of adjustable parameters. Probably the most common way to construct a quadratic classifier is to create a quadratic processor that creates all the terms of $g(X)$, followed by a linear machine which simply weighs each one of these terms.

Figure 21 depicts the implementation of the quadratic gamma detector (QGD) that follows this methodology. The QGD extends the γ CFAR detector as a generalized form by exploiting all quadratic and linear terms of the two intensity features $g_m \bullet X$ and $g_n \bullet X$ in (70).

Notice that in addition to 4 terms of the quadratic form, we included 3 more terms $g_m \bullet X^2$, $g_m \bullet X$, and $g_n \bullet X$ for a direct analogy with the CFAR detector. In fact, our two input features are the intensities at the pixel under test $g_m \bullet X$ and the intensity in the ring neighborhood $g_n \bullet X$. From these quantities the traditional quadratic detector creates $(g_m \bullet X)^2$, $(g_n \bullet X)^2$, $g_m \bullet X$, $g_n \bullet X$, $(g_m \bullet X)(g_n \bullet X)$, and the bias. For a direct comparison with the CFAR we will add $g_m \bullet X^2$ and $g_n \bullet X^2$ for a total of 8 features. The feature vector reads

$$F_{\mu_m, \mu_n} = [g_{m, \mu_m} \bullet X \quad g_{n, \mu_n} \bullet X \quad g_{m, \mu_m} \bullet X^2 \quad g_{n, \mu_n} \bullet X^2 \\ (g_{m, \mu_m} \bullet X)^2 \quad (g_{n, \mu_n} \bullet X)^2 \quad (g_{m, \mu_m} \bullet X)(g_{n, \mu_n} \bullet X) \quad 1] \quad (80)$$

and the quadratic detector becomes

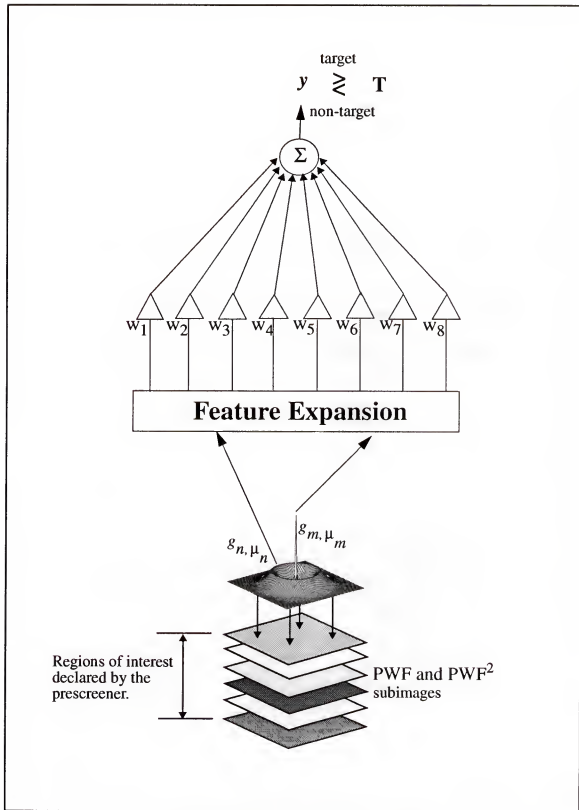


Figure 21 Quadratic Gamma Detector (QGD).

$$\begin{aligned}
y &= \mathbf{W}^T \mathbf{F}_{\mu_m, \mu_n} \\
&= w_1 (g_{m, \mu_m} \bullet X) + w_2 (g_{n, \mu_n} \bullet X) + w_3 (g_{m, \mu_m} \bullet X^2) + w_4 (g_{n, \mu_n} \bullet X^2) \\
&\quad + w_5 (g_{m, \mu_m} \bullet X)^2 + w_6 (g_{n, \mu_n} \bullet X)^2 + w_7 (g_{m, \mu_m} \bullet X) (g_{n, \mu_n} \bullet X) + w_8 \\
&> T \quad (\text{target}) \\
&\leq T \quad (\text{clutter})
\end{aligned} \tag{81}$$

where

$$\mathbf{W} = [w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \quad w_6 \quad w_7 \quad w_8]^T \tag{82}$$

With this formulation, we can understand a little better what was said previously regarding the restricted nature of the two parameter CFAR detector when seen from a pattern recognition stand point. From pattern recognition theory we know that any quadratic or higher order polynomial decision function can be implemented with a linear decision function if the feature vector is appropriately expanded [81]. The added features therefore help increase the separability between target and clutter in pattern space. The weight vector \mathbf{W} is obtained during a training procedure which will be discussed in the following section 4.3.1. The discrimination between the two input classes (target and clutter) is done using a single threshold. The discrimination function is a quadratic function of the image intensity features extracted by the gamma kernels, which leads to naming the detector as the QGD (quadratic gamma detector).

The parameter vector for the γ CFAR (or equivalently for the two parameter CFAR) is

$$\mathbf{W} = [0 \quad 0 \quad 0 \quad -T^2 \quad 1 \quad T^2 \quad -2 \quad 0]^T \tag{83}$$

where some of the parameters are set to zero and others are fixed. Since the increase of the number of free parameters of a system is coupled with more flexibility, we can improve

the performance of the γ CFAR detector by creating more parameters and adapting them with representative data.

4.3.1 Training the QGD

In an adaptive pattern recognition framework, the free parameters of the classifiers which define the positioning of the discriminant function for maximum performance are learned from a set of representative data. This is called the training of the classifier. Given a set of training image chips $\{X_1, X_2, \dots, X_N\}$ centered around points of a known class, we compute the corresponding feature vector F . The corresponding desired values of the chips are 1's for target class and 0's for clutter which construct a desired vector $\mathbf{d} = \{d_1, d_2, \dots, d_N\}^T$.

4.3.1.1 Closed Form Pseudo-Inverse Solutions

If the mean square error between the system output and the desired response is the cost function, there is an analytical solution to the problem [30]. The method solves in the least square sense an overdetermined (assuming $N > 8$) system of linear equations for the unknown coefficient vector \mathbf{W} by using the Moore-Penrose pseudo-inverse.

$$\min_w \| \mathbf{d} - \mathbf{F}\mathbf{W} \|_2 \quad (84)$$

yielding

$$\mathbf{W} = (\mathbf{F}^T \mathbf{F})^{-1} \mathbf{F}^T \mathbf{d} \quad (85)$$

When there exist more weights than equations, the system is under-determined and an optimal weight can be calculated by

$$\mathbf{W} = \mathbf{F}^T (\mathbf{F}\mathbf{F}^T)^{-1} \mathbf{d} \quad (86)$$

In either case, when the pseudo-inverse has zero eigenvalues, singular value decomposition techniques can be used to select the optimal weight vector which has the smallest

Euclidean norm [32].

Since the feature vector F is a function of the parameter μ the problem we are facing is in fact a parametric least squares, which does not have a closed form solution due to the nonlinear dependence on the parameter μ . There are two possibilities for solving this problem. Either we first determine the best values of μ_m and μ_n through an exhaustive search in the parameter space, or we have to use an iterative approach to find both the weight vector, μ_m and μ_n .

One remark is necessary at this point. Training of an adaptive system needs to follow certain steps for good results [1]. In particular the training samples should be representative of the conditions found during the testing, and they should outnumber (at least) 10:1 the number of free parameters in the network. For the QGD the minimum number of training exemplars is easily met. The training and testing procedures are depicted in Figure 22.

The methods of finding a weight vector are further discussed in the following two subsections.

4.3.1.2 Iterative Solution Based on Gradient Descent

An optimal set of μ_m and μ_n can be determined through the parameter space search to provide a minimum false alarm for 100% detection in the training set. However, the search of the parameter space μ is two-dimensional so that finding μ becomes computationally expensive. Computing both the weight vector W and the values of μ_m and μ_n can be accomplished by using a gradient descent method (LMS algorithm), borrowed from adaptive signal processing theory. It is known that the LMS method converges to the Moore-Penrose solution of the overdetermined least squares problem [35].

Using the method of gradient descent, the weights and the parameters are adjusted in an iterative manner along the error surface with the aim of moving them progressively toward the optimum solution. Figure 23 depicts an adaptation scheme for the QGD parameters and weights.

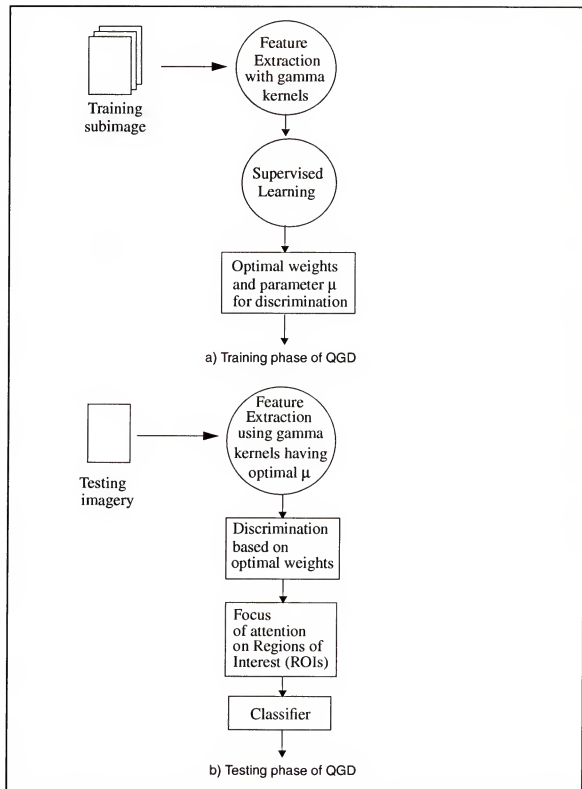


Figure 22 Training and testing of the Quadratic Gamma Detector (QGD).

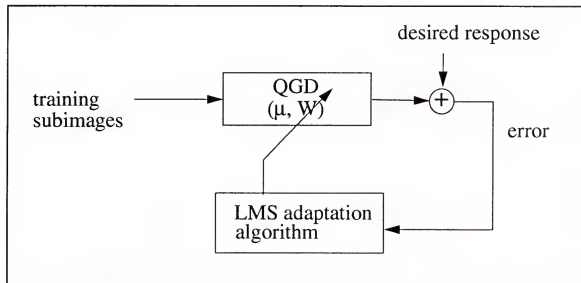


Figure 23 Adaptation scheme for parameters and weights of the QGD.

In the batch mode of adaptation, weight updating is performed after the presentation of all the training examples (this constitutes an epoch). For a given set with N training examples, a cost function can be defined as E_{av}

$$\begin{aligned}
 E(k) &= \frac{1}{p} |d(k) - y(k)|^p \\
 E_{av} &= \frac{1}{N} \sum_{n=1}^N E(k) \\
 &= \frac{1}{Np} \sum_{k=1}^N |d(k) - y(k)|^p
 \end{aligned} \tag{87}$$

where $E(k)$ is the p normed absolute value of an instantaneous error. The adjustment applied to the weights (W) and the parameter (μ) are derived

$$\begin{aligned}
\Delta W &= -\eta \frac{\partial E_{av}}{\partial W} \\
&= -\frac{\eta}{Np} \sum_{k=1}^N \frac{\partial}{\partial W} |d(k) - y(k)|^p \\
&= \frac{\eta}{N} \sum_{k=1}^N \text{sign}(d(k) - y(k)) |d(k) - y(k)|^{p-1} \frac{\partial}{\partial W} y(k) \\
&= \frac{\eta}{N} \sum_{k=1}^N \text{sign}(d(k) - y(k)) |d(k) - y(k)|^{p-1} F_{\mu_1, \mu_{15}}(k)
\end{aligned} \tag{88}$$

$$\begin{aligned}
\Delta \mu_n &= -\alpha \frac{\partial E_{av}}{\partial \mu_n} \\
&= \frac{\alpha}{N} \sum_{k=1}^N \text{sign}(d(k) - y(k)) |d(k) - y(k)|^{p-1} W^T \frac{\partial}{\partial \mu_n} F_{\mu_m, \mu_n}(k) \\
&= \frac{\alpha}{N} W^T \sum_{k=1}^N \text{sign}(d(k) - y(k)) |d(k) - y(k)|^{p-1} \frac{\partial}{\partial \mu_n} F_{\mu_m, \mu_n}(k)
\end{aligned} \tag{89}$$

where

$$\begin{aligned}
\frac{\partial y}{\partial W} &= F_{\mu_m, \mu_n} \\
\frac{\partial y}{\partial \mu_n} &= W^T \frac{\partial F_{\mu_m, \mu_n}}{\partial \mu_n}
\end{aligned} \tag{90}$$

and

$$\text{sign}(\psi) = \begin{cases} 1 & \text{if } \psi > 0 \\ -1 & \text{if } \psi < 0 \end{cases} \tag{91}$$

We can compute the derivative of the feature vector as

$$\frac{\partial F_{\mu_m, \mu_n}}{\partial \mu_m} = \begin{bmatrix} \frac{\partial g_{m, \mu_m} \bullet X}{\partial \mu_m} \\ 0 \\ \frac{\partial g_{m, \mu_m} \bullet X^2}{\partial \mu_m} \\ 0 \\ 2 (g_{m, \mu_m} \bullet X) \left(\frac{\partial g_{m, \mu_m} \bullet X}{\partial \mu_m} \right) \\ 0 \\ (g_{n, \mu_n} \bullet X) \left(\frac{\partial g_{m, \mu_m} \bullet X}{\partial \mu_m} \right) \\ 0 \end{bmatrix}, \quad \frac{\partial F_{\mu_m, \mu_n}}{\partial \mu_n} = \begin{bmatrix} 0 \\ \frac{\partial g_{n, \mu_n} \bullet X}{\partial \mu_n} \\ 0 \\ \frac{\partial g_{n, \mu_n} \bullet X^2}{\partial \mu_n} \\ 0 \\ 2 (g_{n, \mu_n} \bullet X) \left(\frac{\partial g_{n, \mu_n} \bullet X}{\partial \mu_n} \right) \\ (g_{m, \mu_m} \bullet X) \left(\frac{\partial g_{n, \mu_n} \bullet X}{\partial \mu_n} \right) \\ 0 \end{bmatrix} \quad (92)$$

where

$$\begin{aligned} \frac{\partial}{\partial \mu} g_{n, \mu_n} (l_1, l_2) &= \frac{(n+1) \mu_n^n}{2\pi n!} (\sqrt{l_1^2 + l_2^2})^{n-1} e^{-\mu_n \sqrt{l_1^2 + l_2^2}} - \frac{\mu_n^{n+1}}{2\pi n!} (\sqrt{l_1^2 + l_2^2})^n e^{-\mu_n \sqrt{l_1^2 + l_2^2}} \\ &= \frac{(n+1)}{\mu_n} (g_{n, \mu_n} (l_1, l_2) - g_{n+1, \mu} (l_1, l_2)) \end{aligned} \quad (93)$$

The adaptations of W and μ_n can be summarized as

$$\begin{aligned} W(k+1) &= W(k) + \Delta W(k) \\ \mu_n(k+1) &= \mu_n(k) + \Delta \mu_n(k) \end{aligned} \quad (94)$$

One problem of adapting μ is to revisit the 2-D image domain and to run a convolution operation for extracting feature vector F at each iteration. This is computationally intensive. Since the gamma kernels are circularly symmetric, we first project the convolved image to the radial direction, and then use the already developed 1-D algorithm to adapt μ . This will be discussed in details in the following section.

4.3.2 1-D Implementation

The feature vector F is constructed from the convolution of gamma kernels and training subimages. The adaptation of μ requires the extraction of a new feature vector F based on a different kernel shape; that is, we would have to go back to the image plane and compute a new feature vector F after every epoch. The 2-D convolution takes $O(N^d)$ multiplications, so for each training epoch, the adaptation of μ would be extremely computationally expensive even for small training subimages. Fortunately, the 2-D gamma kernels are circularly symmetric so, without loss of information, the result of the convolution can be projected radially to 1-D (see Figure 24) as follows: The pixel intensities in concentric annulus around the pivot point are converted to a 1-D signal, and this signal is then convolved with 1-D gamma kernels.

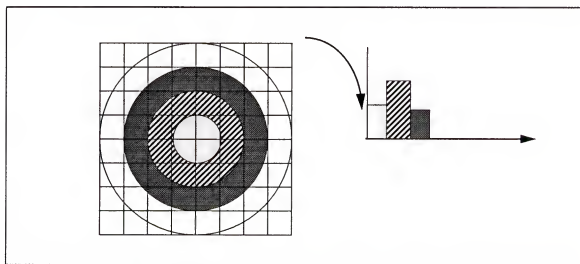


Figure 24 Image converted into 1-D radial energy.

Therefore, we convert the training subimages to 1-D radial profiles, generate 1-D gamma kernels and utilize the recursive implementation of the gamma kernel to perform the convolution. This scheme reduces much of the computational burden of 2-D implementation. The 2-D gamma kernels are reduced to the forms of 1-D gamma kernels.

$$g_{n, \mu}(t) = \frac{2\pi n}{\mu} g_{n, \mu}(l_1, l_2) \Big|_{t = \sqrt{l_1^2 + l_2^2}} \quad (95)$$

The adaptation of the weights W and of μ can be performed using the LMS (least mean square) algorithm in 1-D.

4.4 Comparison of the QGD with the CFAR and the γ CFAR Detectors

To draw a parallel between the QGD decision function (in (81)) and the two parameter CFAR detector (in (51)), we first observe that \bar{X}_c and $\hat{\sigma}_c^2$ are estimates of the local mean and variance, respectively, around the test cell X_t and that the standard deviation can be computed from the first and the second moments by (52) and (53). (54) can be interpreted as a linear decision function with a fixed set of weights for a given T_{CFAR} . The measurements that appear in this equation correspond closely to some of those used in the QGD (in (81)). Specifically, $g_{m, \mu_m} \bullet X$ has the same role as X_t , and $g_{n, \mu_n} \bullet X$ corresponds to the local mean, \bar{X}_c . Similarly, for the second moment we have correspondence between $g_{n, \mu_n} \bullet X^2$ and \bar{X}_c^2 . With this formulation for the QGD we preserved the similarity to the two parameter CFAR detector (the types of features used) but have generalized it with respect to: (1) the shape of the kernels used for the estimation of the mean and variance and (2) the selection of the weights of the decision function which are not chosen a priori but are found through optimization. Note also that the QGD has two additional linear terms and a bias term. The two parameter CFAR detector is therefore a special case of the QGD. In this formulation however, we can no longer guarantee the constant false alarm rate property of the two parameter CFAR which is achieved for Gaussian clutter statistics.

4.5 Artificial Neural Networks (ANNs)

4.5.1 Introduction

Human being performs many complex tasks such as speech and image recognition

with relative ease which are very difficult to solve using traditional algorithmic computing techniques. Much effort has been made on the development of machines which have human-like information processing capabilities. A neural network model, a simplified model of brain, is motivated from the neuroanatomy of living animals with cells corresponding to neurons, activations corresponding to neuronal firing rates, connections corresponding to synapses, and connection weights corresponding to synaptic strengths.

The neural network model is also called a connectionist network and consists of a set of computational units and connection weights between the units, with the processing tasks being distributed across many units. Most of the current neural network models are far from realistic biological neural models but they serve as good models for essential information processing that organisms perform. In this chapter, one of the most popular neural models, a multi-layer perceptron (MLP) model is discussed along with its learning algorithms, validation and generalization problems. The idea of the MLP here is used for extending the QGD into a non-linear structure, which will be later discussed in Chapter 4.6.

4.5.2 Multi-layer Perceptrons (MLPs)

A multi-layer perceptron (MLP) has played a central role in neural network modeling, which is probably the most widely used ANN. An MLP is a feedforward network in which the network input is propagated forward through several processing layers before the network output is obtained. Each layer is composed of a number of nodes, and each node forms a weighted sum of inputs from the nodes in the previous layer and nonlinearly transforms the sum through a bounded, continuously increasing nonlinearity. A multi-layer feed-forward network is shown in Figure 25.

Learning is accomplished by minimizing the error between outputs and target values [68], or by maximizing an entropy measure on the outputs.

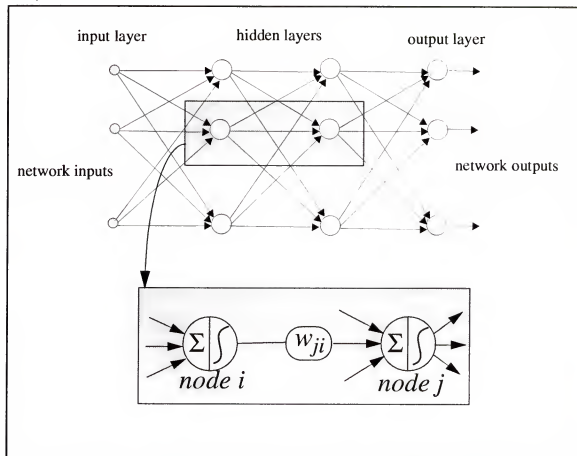


Figure 25 Multi-layer perceptrons (MLPs).

The network learning based on gradient descent requires knowledge of the derivative of the activation functions associated with neurons so that the activation functions need be differentiable with respect to the network weights. The sigmoidal and hyperbolic tangent functions are commonly used as differentiable activation functions in the MLPs. Thus the network output is a continuous (continuously differentiable) function of every weight in the network, enabling it to be trained using gradient descent rules. The availability of such learning algorithms popularized the MLP. The model structure does not depend on the learning rule. It is known that in classification tasks a three-layer MLP network with threshold activation functions could represent an arbitrary decision boundary to arbitrary accuracy [38]. For functional approximation, a three-layer MLP with sufficient nodes in

the hidden layer could approximate, to arbitrary degree, any continuous nonlinear function [37]. It is obvious that an MLP can be used as a tool for providing a general framework for representing non-linear functional mapping between the input space and the output space.

4.5.3 Training MLPs

A neural network has the ability of the network to learn from its environment. A neural network adjusts its free parameters towards minimizing or maximizing a criterion function in an iterative manner during its learning period. There are two general learning paradigms, supervised and unsupervised learning. In supervised learning the output of a network is compared with a desired response and the error between the outputs and the desired responses are used to correct the network parameters. In unsupervised learning any desired responses are not provided to the network. The network discovers for itself interesting categories or features in the input data. In the mid-eighties, a gradient descent learning algorithm called Back Propagation (BP), which enables MLPs to learn arbitrary functional mapping, stimulated considerable interest in learning systems. BP is a supervised learning mechanism in feedforward networks using a cost function and gradient descent which is the most widely used training algorithm. This section discusses BP algorithms (on-line learning and batch learning) for a network having a feed-forward topology and differentiable non-linear activation functions for the case of a differentiable cost function.

4.5.3.1 On-line Learning

In the on-line learning process, a network uses only the information provided by a single training example $\{x(t), y(t)\}$ when the network parameters are adapted. Consider an epoch of N training examples in the following order: $\{x(1), d(1)\}, \dots, \{x(N), d(N)\}$. For a training pattern, each node computes a weighted sum of inputs of the form

$$v_j(n) = \sum_i w_{ji}(n) y_i(n) \quad (96)$$

where $y_i(n)$ is the activation of a node or input, which is connected to node j and $w_{ji}(n)$ is the weight associated with that connection. The sum in Eq 68 is transformed by a non-linear activation function $f(\cdot)$ to give activation $y_j(n)$ of node j in the form

$$y_j(n) = f(v_j(n)) \quad (97)$$

Since the activation outputs are successively computed layer by layer this process is often called forward propagation. Now, the instantaneous error signal at the output of node k at iteration n is defined by

$$e(n) = d_k(n) - y_k(n) \quad (98)$$

On-line back propagation learning generally attempts to minimize the sum of squared instantaneous errors (SSE) at the nodes in the output layer. The SSE is defined by

$$E(n) = \frac{1}{2} \sum_{k=1}^K e(k)^2 \quad (99)$$

where K is the number of nodes in the output layer. For a given training set, $E(n)$ represents the cost function as a measure of training set learning performance. The object of the learning process is to adjust the free parameters of the network so as to minimize the cost function. The adjustments to the weights are made in accordance with the respective errors computed for each training example to the network.

Gradient descent learning algorithms adapt the weights in the direction of the negative gradient of the cost function. The correction $\Delta w_{ji}(n)$ applied to $w_{ji}(n)$ is defined by the delta rule

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)} \quad (100)$$

where η is the learning rate of the BP algorithm. The learning rate controls the speed of network training and affects the network stability. For a node j in the output layer, the par-

tial derivative, $\partial E(n)/\partial w_{ji}(n)$, can be factorized in the form by the chain rule as follows

$$\begin{aligned}\frac{\partial E(n)}{\partial w_{ji}(n)} &= \frac{\partial E(n)}{\partial v_j(n)} \frac{\partial v_j(n)}{\partial w_{ji}(n)} \\ &= -\delta_j(n) \frac{\partial v_j(n)}{\partial w_{ji}(n)}\end{aligned}\quad (101)$$

where $\delta_j(n) = -\partial E(n)/\partial v_j(n)$ is a local gradient at a node in the output layer, which is a sensitive factor of the cost function with respect to the output of a node i before the activation function. Differentiating $v_j(n)$ with respect to $w_{ji}(n)$ yields

$$\frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n) \quad (102)$$

The local gradient $\delta_j(n)$ at a node j in the output layer is obtained by

$$\delta_j(n) = e_j(n) \varphi'(v_j(n)) \quad (103)$$

Accordingly, the correction $\Delta w_{ji}(n)$ with (102) and (103) can be written by

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) \quad \text{at a node } j \text{ in the output layer} \quad (104)$$

For a node in a hidden layer, the local gradient can be computed recursively in terms of the local gradients of all nodes in the next layer in the forward direction.

$$\begin{aligned}\delta_j(n) &= -\frac{\partial E(n)}{\partial v_j(n)} \\ &= \sum_k -\frac{\partial E(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial v_j(n)} \\ &= \sum_k \delta_k(n) \frac{\partial v_k(n)}{\partial v_j(n)}\end{aligned}\quad \text{at a node } j \text{ in a hidden layer} \quad (105)$$

where the sum runs over all units k to which the node j sends connections. The net activation level at node k is

$$v_k(n) = \sum_j w_{kj}(n) y_j(n) \quad (106)$$

The partial derivative in (105) can be rewritten

$$\begin{aligned}\frac{\partial y_k(n)}{\partial v_j(n)} &= \frac{\partial y_k(n)}{\partial v_k(n)} \frac{\partial v_k(n)}{\partial v_j(n)} \\ &= \sigma'(v_k(n)) w_{kj}(n)\end{aligned}\quad (107)$$

Thus, using (105) and (106), we obtain the following back-propagation formula

$$\delta_j(n) = \sigma'(v_j(n)) \sum_k \delta_j(n) w_{kj}(n) \quad \text{at a node } j \text{ in a hidden layer} \quad (108)$$

The local gradient at a node in the j th hidden layer can be computed by propagating the local gradients backwards from the nodes in the $j+1$ layer.

4.5.3.2 Learning Rate and Momentum

The network training based on the gradient descent method requires to choose a suitable value for the learning rate η . The effectiveness and convergence of the BP learning algorithm depend significantly on the value of the learning rate η . When the curvature of the error surface varies significantly with the direction of interest in the weight space a large value of η will result in oscillation in the error surface and, for a fairly flat error surface, a small value of η will result in a very slow convergence in the error minimization process because the weight adaptation applied to a weight is proportional to the derivative of E with respect to the weight. Only small learning rates guarantee a true gradient descent, increasing the total number of learning steps that needed to reach a satisfactory solution. One of the simplest methods in accelerating convergence speech is addition of a momentum term in the weight adaptation (in (104)).

$$\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n) + \alpha \Delta w_{ji}(n-1) \quad (109)$$

where α is a momentum rate and $\Delta w_{ji}(n-1)$ is a weight correction when the $n-1$ th training example is present at the network input.

Summary of Back Propagation Algorithm

1. Determine a network topology, initialize randomly all weights and set a learning rate and momentum rate.
2. Repeat until a network performance is satisfied.
 - 2.1 Forward Propagation step: Feed-forward the training examples into the network and compute the activations of all nodes
 - 2.2 Calculate errors in the output layer and compute the local gradients at all nodes

$$\delta_j(n) = e_j(n) \sigma'(v_j(n)) \quad \text{at node } j \text{ in the output layer} \quad (110)$$

$$\delta_j(n) = \sigma'(v_j(n)) \sum_k \delta_j(n) w_{kj}(n) \quad \text{at node in a hidden layer} \quad (111)$$

- 2.3 Update the weights

$$w_{ji}(n+1) = w_{ji}(n) + \eta \delta_j(n) y_i(n) + \alpha \Delta w_{ji}(n-1) \quad (112)$$

4.5.3.3 Batch Learning

In on-line BP learning, the weight adaptation was performed after the presentation of each training example. The local gradient estimates use only a single piece of training information when the weight adaptation is performed.

In batch BP learning, the weights are adjusted after presenting the entire set of training examples. The batch learning provides a more accurate estimate of the gradient vector. The instantaneous squared errors are summed up over the entire training examples and the average of the total squared errors can be defined as a cost function

$$E_{av} = \frac{1}{2N} \sum_{n=1}^N \sum_{k=1}^K e(k)^2 \quad (113)$$

where N is the number of training examples. A correction Δw_{ji} applied to a weight w_{ji} is proportional to the negative gradient of the cost function

$$\begin{aligned}
 w_{ji} &= -\eta \frac{\partial E_{av}}{\partial w_{ji}} \\
 &= -\frac{\eta}{N} \sum_{n=1}^N e_j(n) \frac{\partial e_j(n)}{\partial w_{ji}}
 \end{aligned}
 \tag{114}$$

4.5.4 Validation of A Neural Model

In any learning system, the validation of system learning is one of the most important parts because the objective of network training is not to learn an exact representation of the training data itself, but rather to learn the underlying statistics of the process which govern the data. Therefore, during the learning phase, the network should assess how well it has learned the training data and is able to generalize unforeseen data. In order to develop quantitative techniques to evaluate a network's performance with real-world data, rigorous mathematical foundations must be developed to determine the characteristics of the training set and the network's ability to generalize from the training set. However, although the question of whether a network possesses the ability to generalize correctly (or sufficiently accurately) is still unsolved, most learning algorithms can successfully learn a set of training examples given a sufficiently flexible model structure or an appropriate learning algorithm. A popular methods in assessing the validity of trained networks is to split the available data into two sets, a training set and a test set. The training set is further split into two subsets; one set used for training the network and the other for evaluating the performance of the network during the training phase [80].

4.5.4.1 Bias/Variance Dilemma

A network model that is too inflexible will have a large bias, while one that is too flexible will have a large variance. This can be explained by the bias/variance dilemma [28]. The MSE performance measure for a given training set can be decomposed into the sum of

two components which reflect the squared bias of the network error, and the variance of the estimates of the trained network.

Let D denote a training set, $y(x,D)$ be the output of a network trained using the data contained in D and $E_D(\cdot)$ denote the expectation operator taken over all possible training sets, then the output error is given by

$$e_y(x) = \hat{y}(x) - y(x, D) \quad (115)$$

The effectiveness of the network as a predictor of $\hat{y}(x)$ by calculating the MSE for all possible training sets D is given by

$$\begin{aligned} E_D(e_y(x)) &= E_D(\hat{y}(x) - y(x, D)) \\ &= (\hat{y}(x) - E_D(y(x, D)))^2 + E_D((\hat{y}(x) - y(x, D))^2) \end{aligned} \quad (116)$$

The first term in the right hand side is called the square of the bias, and the second term estimates the variance of the network approximations. The bias measures the average modelling error and the variance measures how sensitive the network modelling is to a particular choice of data set. When the network is too flexible, a large variance occurs so that the performance of the network is very sensitive to a particular training set. This results in a poor MSE performance. The network also produces a poor MSE performance when it possesses too little flexibility which causes a large bias. Bias and variance are complementary quantities. In general, a network should be flexible enough to ensure that the modelling error (bias) is small but should not be over-parameterized because its performance is highly sensitive to a particular training set (high variance).

4.5.4.2 Network Complexity and Early Stopping

The problem of choosing a network complexity (in an MLP) is to choose the size of the free parameter set used to model the data set (in terms of number of nodes and hidden layers). A good estimate of the true performance of a network is required to determine

whether the complexity of a network model is effective for a particular data set. If the model complexity is increased and this resulted in a lower modelling error, it would indicate that the network is not flexible. Similarly, if a more flexible model produced a higher modelling error, this would indicate that the network overfitted the training data set and a simpler network would be preferred for a better generalization.

An alternative to obtaining the effective complexity of a network is the procedure of early stopping. Typically, iterative training of learning systems reduces a cost function error as more iterations are made during the training phase. However, the generalization error measured with respect to a validation set often decreases as the number of iterations increases and starts increasing after a certain iteration point where the training causes the network to over-fit the training set. For a good generalization, training can be stopped at the point the generalization error starts increasing. This is referred to as the cross-validation and this procedure provides an appealing guide to better generalization performance of the network.

4.6 Nonlinear Extension To QGD (NL-QGD)

4.6.1 Introduction

One of the possible extensions to the QGD is to augment the output adder with a set of nonlinear processing elements which nonlinearly combine the feature elements, i.e. to implement a neural classifier such as an MLP. The structure is called the NL-QGD (Non-Linear extension to QGD). Since the MLP is capable of creating arbitrary discriminant functions this extension has the potential to improve performance. Note that the QGD creates a quadratic discriminant function of the image intensity, which is only optimal for unimodal probability density functions [8] [27]. Moreover, a nonlinear system normally generalizes better, so the performance in the test set can also improve.

In order to fully use the neural network approach we have to develop an iterative algo-

rithm to adapt both the weights and μ (using the backpropagation procedure). Availability of on-line methods of adapting μ effectively means that the detector output error can be used for refining μ , or equivalently, to search on-line for the optimal guard band. The relationship between the minimum number of false alarms and the minimum mean square error becomes crucial now, since the system will be adapting μ for the smallest MSE. The cost function used should correspond to the minimum false alarm rate, since this is the basis of scoring. Alternate error norms may have to be used to adapt the weights to guarantee that the minimum cost corresponds to the minimum false alarm rate.

Figure 26 displays the block diagram of the NL-QGD. As indicated in this figure the feature expansion is still the same as that of the QGD. In Figure 21, we can think of the QGD as the linear part of one of the hidden layer processing elements. The two basic issues that need to be addressed when using a neural network are the training and the net topology.

4.6.2 Training the NL-QGD

In order to fully develop the neural network approach, an iterative learning scheme is required to adapt the weights and the parameter μ . Availability of on-line methods of adapting μ effectively means that the detector output error can be used to search on-line for the optimal local area and guard band.

The NL-QGD is trained with a desired signal d (1s for the target class and 0s for the non-target class) using a back-propagation algorithm [68].

$$d(n) = \begin{cases} 1 & \text{feature vector belongs to the target class} \\ 0 & \text{feature vector belongs to the non-target class} \end{cases} \quad (117)$$

The sum of squared errors is utilized in this work, i.e.

$$\min_W \left(E(n) = \frac{1}{2N} \sum_{n=1}^N \|d(n) - z(n)\|^2 \right) \quad (118)$$

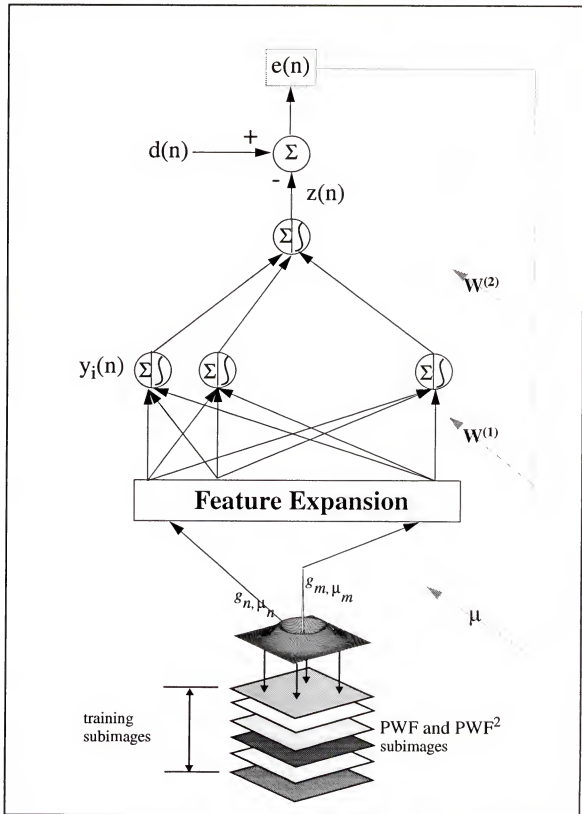


Figure 26 Adaptation scheme of the NL-QGD

Moreover, in order to adapt the parameter μ which controls the scale of the γ CFAR stencil, the error generated at the detector output is back-propagated up to the input layer. The decision boundary of the NL-QGD is therefore formed in conjunction with the parameter μ . The correction $\Delta\mu(n)$ at each iteration is proportional to the instantaneous gradient $\partial E(n)/\partial\mu(n)$. According to the chain rule, this gradient is expressed as follow,

$$\frac{\partial E(n)}{\partial\mu(n)} = \sum_i \frac{\partial E(n)}{\partial v_i(n)} \frac{\partial v_i(n)}{\partial\mu(n)} \quad (119)$$

where the index i runs over the nodes in the first hidden layer. $v_i(n)$ is the output before the nonlinear transformation is applied at the i th node in the first hidden layer and is expressed by

$$v_i(n) = \sum_{p=1}^P w_{ip}(n) f_p(n) \quad (120)$$

where the feature element $f_p(n)$ of F is given by (80) at each iteration n and

$$F(n) = [f_1(n), f_2(n), \dots, f_P(n)]^T \text{ with } P = 8.$$

The local gradients $\partial E(n)/\partial v_i(n)$ can be computed by (105). Therefore, (119) is written with the partial derivative of $v_i(n)$ with respect to $\mu(n)$ as

$$\begin{aligned} \frac{\partial E(n)}{\partial\mu(n)} &= \sum_i \delta_i(n) \sum_{p=1}^P w_{ip}(n) \frac{\partial f_p(n)}{\partial\mu(n)} \\ &= \sum_{p=1}^P \sum_i \delta_i(n) w_{ip}(n) \frac{\partial f_p(n)}{\partial\mu(n)} \\ &= \sum_{p=1}^P \delta_p(n) \frac{\partial f_p(n)}{\partial\mu(n)} \\ &= \Delta_P^T(n) \frac{\partial F(n)}{\partial\mu(n)} \end{aligned} \quad (121)$$

where $\delta_i(n)$ and $\delta_p(n)$ are the local gradients at the i th node in the first hidden layer and p th node in the input layer respectively. The local gradient vector in the input layer is defined as $\Delta_P(n) = [\delta_1(n) \ \delta_2(n) \ \dots \ \delta_P(n)]^T$. $w_{ip}(n)$ is a weight between the p th

node in the input layer and the i th node in the first hidden layer. The gradient $\partial F(n) / \partial \mu(n)$ is given by (92).

The adaptation of the parameter μ is given by

$$\mu_j(n+1) = \mu_j(n) - \beta \frac{\partial E(n)}{\partial \mu_j(n)} = \mu(n) + \beta \Delta_p^T(n) \frac{\partial}{\partial \mu_j(n)} F_{\mu_1, \mu_{15}}(n) \quad (122)$$

where β is the step size, and $j = n, m$ (one equation for each kernel).

CHAPTER 5

TRAINING STRATEGIES FOR NL-QGD

5.1 Introduction

Neural network models are usually defined by three major parts: architecture, cost function and training algorithm. Here the cost function typically measures errors between network estimates and the actual outputs from the training data and plays an important role as a guidance for network optimization.

The sum of squares error (SSE) function (L_2 norm) has been widely used as an optimality index for network training. There are many other possible choices of cost functions which can be also considered, depending on particular applications. Minimizing the SSE is equivalent to the maximum likelihood principle for Gaussian distribution errors in a regression problems where the goal is to model the conditional distribution of the output variables given the input samples.

When used as a classification device, a feed-forward neural network is normally trained with binary desired responses (0 or 1) given training data during the training period. In order to guarantee that the outputs represent probabilities, the sum of the output values should be equal to one and each output value should lie in the range (0, 1). For classification problems, the error distribution is far away from the Gaussian distribution because the target variables are binary. For two-class classification problems, the error has the form of a binomial distribution and multinomial distribution for multi-class classification problems in which the network has the same number of output nodes with the number of class membership.

Cross-entropy can also be used as a cost function for network optimization [2] [36] [79]. The cross-entropy cost function maximizes the likelihood of observing a training data set during the training. In this sense, the cross-entropy is theoretically the most appropriate cost function for network optimization in the case of binary classification [69].

However, when a neural network is used as a detector with a single output node in the output layer for two classes and supposed to detect samples from one particular class and reject samples from other classes as much as possible, the network needs to be trained on an optimality condition which is different from the classification problems. Presently this optimal condition has not been formulated. So the problem has to be solved experimentally. Through the search of cost functions for network optimization, the performance of the two detectors (QGD and NL-QGD) are discussed based on cost functions used for training.

5.2 Optimality Index

5.2.1 L_2 Norm

One of the most popular norms in training networks is the L_2 criterion. For a network with linear outputs, training the network leads to an optimal solution, the least squares solution. The L_2 norm is an appropriate choice for normally distributed inputs in the sense of both minimum cost and minimum probability of prediction error (maximum likelihood). Consider a set of training pairs $\{x(n), d(n)\}$, $n = 1, 2, \dots, N$. Assume that the input vectors x_i are drawn randomly and independently from a normal distribution. The objective of network training is not to memorize training data, but rather to model the underlying generator of the data. Therefore the best possible prediction for desired responses d can be made when new data is presented to the trained network. A network training scheme is shown in Figure 27. The most general and complete description of the generator of the data is in terms of the probability density $p(x, d)$ in the joint input-target space [5].

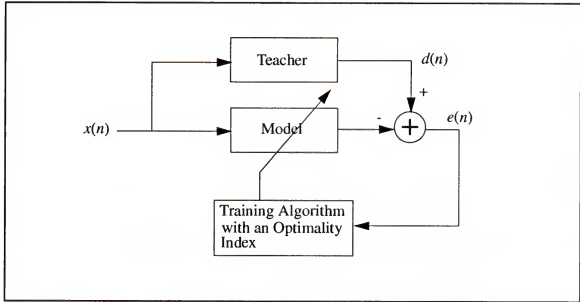


Figure 27 A network training scheme.

The joint probability of the input and desired responses can be expressed in terms of the conditional probability density of the target data and unconditional probability density of the input data,

$$p(x, d) = p(x|d)p(x) \quad (123)$$

where $p(x|d)$ is the probability density of the target variable given an input x , and $p(x)$ is the probability density of the input given by

$$p(x) = \int p(x, s) ds \quad (124)$$

By the principle of maximum likelihood, we want to maximize the joint probability density in the input-target space given a set of input and target data. It is often convenient to minimize the negative logarithm of the likelihood, $p(x, d)$ since the negative logarithm is a monotonic function. We therefore minimize

$$\begin{aligned} E &= -\ln p(x, d) \\ &= -\sum_n \ln p(d(n)|x(n)) - \sum_n \ln p(x(n)) \end{aligned} \quad (125)$$

Since the second term in (125) is independent of the network parameter, minimizing E is equivalent to minimizing the first term so the error function can be written as

$$E = -\sum_n \ln p(d(n)|x(n)) \quad (126)$$

We assume that the input data has a Gaussian distribution, the target value has a form of deterministic function with added Gaussian noise e_k , so that

$$d_k = h_k(\mathbf{x};\mathbf{w}) + e_k \quad (127)$$

We now want to model the deterministic function $h_k(\mathbf{x})$ by a neural network with output $y_k(\mathbf{x};\mathbf{w})$ where \mathbf{w} is the set of weight parameters determining the network mapping. The distribution of $e(n)$ is given by

$$p(e_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{e_k^2}{2\sigma^2}\right) \quad (128)$$

Using (127) and (128), the probability density of the target variables is given by

$$p(d_k|x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y_k(\mathbf{x};\mathbf{w}) - d_k)^2}{2\sigma^2}\right) \quad (129)$$

The mapping function between input and target, $h_k(\mathbf{x})$, was replaced by the network model $y_k(\mathbf{x};\mathbf{w})$. Substituting (129) into (126), we obtain the error function [5],

$$E = \frac{1}{2\sigma^2} \sum_{n=1}^N \sum_{k=1}^C \{y_k(x(n);w) - d_k(n)\}^2 + NC \ln \sigma + \frac{NC}{2} \ln(2\pi) \quad (130)$$

where C is the number of classes and N the number of exemplars. In (130), the second and third term are independent of the network parameters \mathbf{w} and can be omitted. The pre-multiplication factor in the first term can also be omitted. The error function then has a form of the SSE function

$$E = \frac{1}{2} \sum_{n=1}^N \sum_{k=1}^C \{y_k(x(n);w) - d_k(n)\}^2 \quad (131)$$

Here, the SSE cost function was derived from the principle of maximum likelihood on the assumption of Gaussian distributed target data.

5.2.2 Training with Excluding Outliers from Non-Target Class with L_2 Norm

In the training data for the QGD and NL-QGD, it is observed that some of the target and non-target class data overlap because when a pixel intensity significantly exceeds the mean intensity in its surrounding region the two-parameter CFAR or γ CFAR detector declares the pixel as detected because targets and man-made objects (non-targets) are usually brighter than background in their surrounding areas.

The objective of training the NL-QGD (or any detector for that matter) is not to obtain minimum classification error but to achieve a minimum false alarm rate while maintaining a high probability of target detection. The training of neural networks are mostly intended to produce minimum classification error in given training data and to well generalize the data which has never been seen by the network. This leads to partitioning of the input space into subregions according to the number of classes and usually places a decision boundary among highly populated regions of classes.

In two-class detection problems, the input space is partitioned into two subspaces, one for the target class and the other for the non-target class. The NL-QGD produces very low values of outputs for target data which are far away from the decision boundary and reside in a deep region of the non-target subregion. These low output values really degrade high probability of target detection because a threshold for the NL-QGD is set based on minimum target output value for a 100% target detection. It is therefore desired that a decision boundary be placed in order to encompass all target samples into the target subregion and exclude non-target samples by as many as possible from the target subregion. This, of course, may not yield minimum classification errors but leads to obtaining high probability detection in favor of one class.

Figure 28 illustrates the effect of outliers in placing a decision boundary in the input

space. In Figure 28a, the data set contains 15 samples (marked by “*”) for target class and 15 samples (marked by “o”) in the input space. The desired responses are 1’s and -1’s respectively for the target class and non-target class.

In Figure 28a, the decision boundary #1 was configured based on the LS solution by using all samples while the decision boundary #2 was formed by the LS solution which was computed after removing four non-target outliers ((-3,1), (-2,2), (-1,2) and (-1 1)) from the data set. By removing the four non-target outliers when computing a LS solution, the decision boundary moves in the direction where the target outliers are located so that more target-class samples are included above the decision boundary #2. Figure 28b plots the outputs of the data set based on the two LS solutions. From the decision boundary #1, 10 false alarms occurs in order to detect all target outputs. However the decision boundary #2 yields 8 false alarms for all target detection.

There is a lack of theoretical foundation to design cost functions for network training in such a case. A simple way of implementing the idea is to train the NL-QGD, excluding outliers from non-target samples during the training period. Since outliers from the training data cause low output values for target class and high output values for non-target class, this yields large error values which are used to correct network parameters during training. Error values serve as forces that position the decision boundaries in the direction of the corresponding samples. Therefore removing non-target outliers helps the decision boundary to shift towards the direction of target outliers. This can yield larger output values of the NL-QGD from target outliers and reduces false alarms correspondingly on the operating point of high probability target detection.

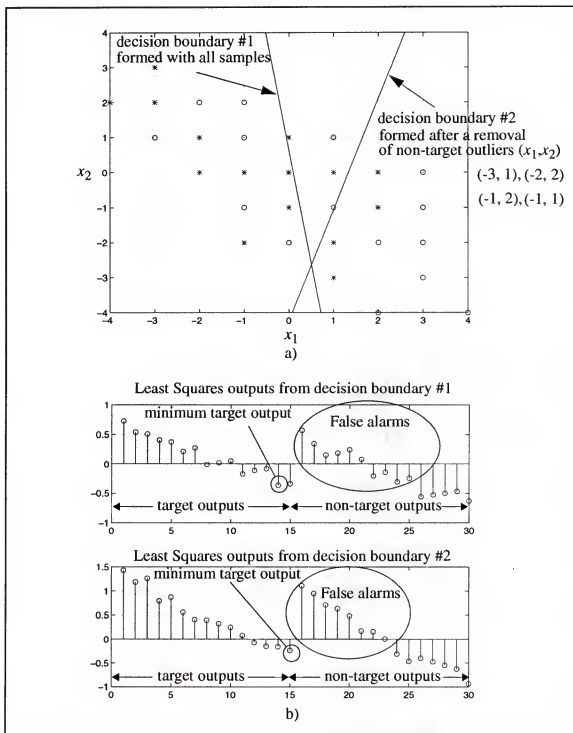


Figure 28 An example of outlier effect to decision boundary formation and false alarm rate.

Training procedure of NL-QGD with removal of non-target outliers

1. Train NL-QGD with a complete set of training data and cross validation data for a certain number (N_1) of iterations
2. Compute false alarm for 100% target detection from the cross validation set.
 - 2.1 *if* ($\min FA > FA(\text{iter})$) {
 - $\min FA = FA(\text{iter})$
 - $\text{stopiter} = \text{stopiter} + 1$
 - remove the non-target exemplar which caused the largest output values from the non- target exemplars of the training data.
 - }
 - else if* ($\min FA = FA(\text{iter})$) {
 - if* ($\text{stopiter} < N_2$) $\text{stopiter} = \text{stopiter} + 1$
 - else* stop training
 - }
 - 2.2 train NL-QGD for a certain number of iterations (N_3).
 - 2.3 go to step 2

5.2.3 L_p Norm

The L_2 norm (SSE function) was derived from the Gaussian distributed target variables. The L_2 norm equally weighs all errors. When there are long tails on the distributions then the solution can be significantly affected by a very small number of points called outliers which cause particularly large errors. We can obtain more general error functions by extending the Gaussian distribution function to a more general form [5] [7]

$$p(e) = \frac{p^{1-1/p}}{2\sigma_p \Gamma(1/p)} \exp\left(-\frac{|e|^p}{p(\sigma_p)^p}\right) \quad (132)$$

where $\Gamma(\cdot)$ denotes the Gamma function and σ_p is the dispersion parameter in the L_p sense. For the case of $p = 2$ this reduces to a Gaussian. Substituting (132) into (125) and omitting constant terms, we obtain a generalized error function of the SSE

$$E = \sum_{n=1}^N \sum_{k=1}^C |d_k(n) - y_k(x(n); w)|^p \quad (133)$$

This shows the link between error distribution and L_p norm criterion function. Figure 29 shows error weighting effect of L_p norms according to a norm power, p . When p increases, large errors become more weighed than small errors. On the other hand, small error get more weighed than large errors as p decreases below $p = 1$. This implicates that the decision boundary forms differently in the input space, depending upon norms.

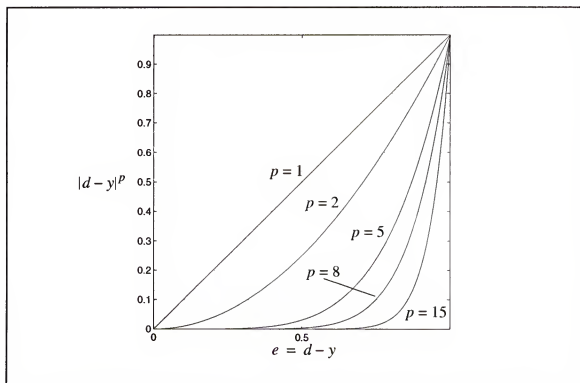


Figure 29 $|d - y|^p$ versus e for different p .

The derivatives of the L_p norm error function with respect to the network parameters are [5] given by

$$\frac{\partial E}{\partial w_{ji}} = \sum_n \sum_k |d_k(n) - y_k(x(n); w)|^{p-1} \times \text{sign}(d_k(n) - y_k(x(n); w)) \frac{\partial}{\partial w_{ji}} y_k(x(n); w) \quad (134)$$

If we view the L_p norm error function in an asymptotic sense the error function is

$$\begin{aligned}
\lim_{N \rightarrow \infty} \frac{1}{N} \sum_{n=1}^N \sum_{k=1}^C |d_k(n) - y_k(x(n); w)|^p \\
= \sum_{k=1}^C P(\omega_k) \int |d_k - y_k(x; w)|^p p(x | \omega_k) dx \\
= \sum_{k=1}^C P(\omega_k) \int |d_k - y_k(x; w)|^2 f(x) dx
\end{aligned} \tag{135}$$

where

$$f(x) = |d_k - y_k(x; w)|^{p-2} p(x | \omega_k) \tag{136}$$

For $p = 2$, the cost function (L_2) places emphasis on the errors from samples in densely populated regions, that is, in the regions where $p(x)$ is large, rather than in the regions near the decision boundary. When p is large the optimization process of the network is dominated by the samples which cause large errors. If p is taken to be small, more emphasis is placed in the region of the decision boundary.

A BP algorithm for MLPs can be used with only minor modification at the local gradient in the output layer. The local gradients in the output layer is modified as

$$\begin{aligned}
\delta_l(n) &= \frac{\partial}{\partial v_l} E(n) \\
&= \phi'(y_l(n)) \sum_{i=1}^L \text{sign}(d_i(n) - y_i(n)) |d_i(n) - y_i(n)|^{p-1}
\end{aligned} \tag{137}$$

where l and L are a node index and the number of nodes in the output layer respectively. $E(n)$ is a instantaneous error defined as

$$E(n) = \frac{1}{p} \sum_{l=1}^L |d_l(n) - y_l(n)|^p \tag{138}$$

Large norms usually slows down the training speed. This effect can be seen by rewriting (137) in the following equation

$$\delta_l(n) = |e(n)|^{p-2} \varphi'(y_l(n)) \sum_{l=1}^L \text{sign}(d_l(n) - y_l(n)) |e(n)| \quad (139)$$

The weight adaptation is

$$\begin{aligned} \Delta w_l(n) &= \eta \delta_l(n) y_l(n) + \alpha \Delta w_l(n-1) \\ &= \eta_p \varphi'(y_l(n)) y_l(n) \sum_{l=1}^L \text{sign}(d_l(n) - y_l(n)) |e(n)| \\ &\quad + \alpha \Delta w_l(n-1) \end{aligned} \quad (140)$$

where η_p is the effective learning rate for a norm (p) and $\eta_p = \eta |e(n)|^{p-2}$. When $|e(n)| < 1$, η_p become very small for large norms ($p > 2$) so that the convergence speed becomes very slow. Note that η is the learning rate for $p = 2$.

5.2.4 Mixed L_p Norm

The errors from the training data are equally penalized with the L_2 norm. For $p < 2$, more emphasis is placed on smaller errors than on large errors while the L_p norms ($p > 2$) penalize larger errors more than smaller errors. So, by using mixed norms, the different emphasis on the target class samples and the non-target class samples can be made in training the NL-QGD. It is desirable to use a larger norm ($p > 2$) for the target class in order to prevent the NL-QGD from producing very low output values from target outliers and a smaller norm ($p < 2$) for the non-target class. Since the smaller norm weighs the smaller errors more, large errors from non-target outliers could not gain force as much as the L_2 case to pull the decision boundary into their directions. A cost function can thus be defined with a mixed form of two different norms,

$$E = \left[\frac{1}{N_{nt}} \sum_{X \in \omega_1} |d - y(X, w)|^{pl} \right]^{1/pl} + \left[\frac{1}{N_n} \sum_{X \in \omega_2} |d - y(X, w)|^{ph} \right]^{1/ph} \quad (141)$$

where N_{nt} is the number of non-target samples and N_t is the number of target samples. pl is a small norm ($pl < 2$) and ph is a large norm. In order to prevent the sum of pl -power errors from dominating the sum of ph -power errors ($ph > 2$) when errors are less than 1 (the case

for NL-QGD with a sigmoidal activation function at the output) those sums are normalized by the numbers of their class samples and the power of the inverses of pl and ph respectively. The weight adaptation is performed after presenting all the samples each time (batch training). The local gradient at the output can be written as

$$\begin{aligned} \delta_k &= \frac{\partial E}{\partial v_k} \\ &= \left[\frac{1}{N_{nlX \in \omega_1}} \sum |d - y(X, w)|^{pl} \right]^{\frac{1}{pl} - 1} \frac{1}{N_{nlX \in \omega_1}} \sum |d - y(X, w)|^{pl-1} \varphi'(v_k) \\ &\quad + \left[\frac{1}{N_{nX \in 2}} \sum |d - y(X, w)|^{ph} \right]^{\frac{1}{ph} - 1} \frac{1}{N_{nX \in 2}} \sum |d - y(X, w)|^{ph-1} \varphi'(v_k) \end{aligned} \quad (142)$$

5.2.5 Cross Entropy

The binomial distribution, one of the most useful discrete distributions, is based on the idea of a Bernoulli trial [69]. A Bernoulli trial is an experiment with only two possible outcomes. Observation of this nature arise, for instance, in medical trials where, at the end of the trial period, a patient has either recovered ($d = 1$) or has not ($d = 0$). When a network uses a single output which can meet probability conditions for two-class classification problems, we want the value of the output, y , to represent the posteriori probability $p(\omega_1|x)$ for class ω_1 , that is, $y = p(\omega_1|x)$ and for class ω_2 , $p(\omega_2|x) = 1 - y$. In our problem, $y = P(\text{target class/input image chip})$ for the target class ($d = 1$) and $1 - y = P(\text{non-target/input image chips})$ for the non-target class ($d = 0$). So we combine this scheme into a single scheme. The probability of observing either target or non-target is

$$p(d|x) = y^d (1 - y)^{1-d} \quad (143)$$

This is the Bernoulli distribution which is a special case of the binomial distribution. Assuming that the target or non-target data are drawn independently from this distribution,

the likelihood function ($p(d|x)$) is expressed by

$$\prod_n y(n)^{d(n)} (1 - y(n))^{1 - d(n)} \quad (144)$$

The goal is to maximize this likelihood function given input data. For convenience, we again want to minimize the negative logarithm of the likelihood function which can be thought as a cost function (E) for the network optimization [5] [69].

$$E = -\sum_n \{ d(n) \ln y(n) + (1 - d(n)) \ln (1 - y(n)) \} \quad (145)$$

This error function is called the cross-entropy error function between desired responses (d) and the posterior probabilities (y). This shows that the cross entropy is a natural cost function for the two class classification. Now, the derivative of this function is

$$\frac{\partial E}{\partial y(n)} = \frac{y(n) - d(n)}{y(n) (1 - y(n))} \quad (146)$$

Note that when $y(n) = d(n)$ for all n $E = 0$. With the interpretation of output activations having probabilities, we want them to range between 0 and 1. Therefore a sigmoidal function is natural and the posterior probability can be written in the form of logistic function.

$$p = \frac{1}{1 + e^{-\eta}} \quad (147)$$

We see that the derivative of E with respect to e is

$$\frac{\partial E}{\partial \eta} = y(n) - d(n) \quad (148)$$

The error form has the same form of sum of squared error with linear output units. But the outputs here represents the posterior probabilities. η can be a linear discriminant function or a nonlinear discriminant function.

For desired responses having 1 for target class and 0 for non-target class, the cross entropy cost function can be written in the following form [5]

$$E = - \sum_{n \in \text{target class}} \ln(1 - e(n)) - \sum_{n \in \text{non-target class}} \ln(1 + e(n)) \quad (149)$$

where $e(n) = d(n) - y(n)$. For small $e(n)$, the cross entropy function becomes

$$E \approx \sum_n |e(n)| \quad (150)$$

This has the same form of L_1 norm. Small errors are more weighed by the cross entropy function than L_p norms ($p > 2$). The cross entropy cost function versus error is plotted in Figure 30. The small errors and large errors are penalized more by the cross entropy cost function than L_p norms.

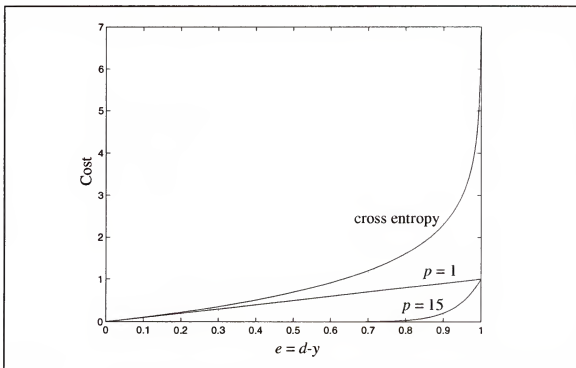


Figure 30 Cross entropy cost function versus error.

A BP algorithm can be used for training MLPs with the cross entropy cost function. The local gradients are modified in the output layer of MLPs and given by

$$\delta_l = \frac{\partial}{\partial v_l} E(n)$$

$$= \begin{cases} \sum_{l=1}^L \frac{1}{1 - e_l(n)} \varphi'(v_l(n)) & \text{if } n \in \text{target class} \\ \sum_{l=1}^L \frac{1}{1 + e_l(n)} \varphi'(v_l(n)) & \text{if } n \in \text{non-target class} \end{cases} \quad (151)$$

where $E(n)$ is an instantaneous error defined as

$$E(n) = \begin{cases} -\sum_{l=1}^L \ln(1 - e(n)) & \text{if } n \in \text{target class} \\ -\sum_{l=1}^L \ln(1 + e(n)) & \text{if } n \in \text{non-target class} \end{cases} \quad (152)$$

CHAPTER 6

EXPERIMENTS AND RESULTS

6.1 Introduction

This chapter experiments the focus of attention proposed for ATR and evaluates the system performance based on a millimeter wave SAR imagery (Mission 90 Pass 5 SAR data set).

In Section 6.2, the two prescreeners, the two-parameter CFAR detector and the γ CFAR detector, are evaluated based on their detection performances over the entire imagery of the Mission 90 Pass 5 SAR data set. Section 6.3 assesses the detection performance of the QGD and the NL-QGDs applied to the regions of interest (ROIs) passed by the two-parameter CFAR detector. The results of discriminating powers of the QGD and NL-QGDs are compared based on ROCs over the false alarms and targets detected by the two-parameter CFAR detector. The detection performance of the NL-QGDs trained based on different optimality indices are presented for different sizes of networks. The detection performances of the QGD and NL-QGDs are presented in conjunction with the γ CFAR detector in Section 6.3.4.

6.2 Prescreening SAR Imagery

6.2.1 Two-Parameter CFAR Processing

The two-parameter CFAR detector was run over the 127 frames (about 7 km^2) of the mission 90 pass 5 data. 345 targets from the TABILS 24 ISAR data base were embedded based on the method mentioned in Section 2.7.2. The size of the CFAR stencil was 85 by

85 pixels in order to compare with the CFAR stencil (84 by 84 pixels for 1 ft resolution PWF SAR data) by Novak et al. [57]. The local mean and standard deviation are computed in the outmost 4-pixel wide boundary of the stencil. The intensity of the test pixel is computed by averaging 3 by 3 pixels in the center of the stencil.

After the CFAR processing, multiple detection points occur in targets and other regions because targets and man-made clutter (and tree tops) normally consist of many high reflectivity pixels that trigger the prescener repeatedly (raw detections). It is therefore required that a clustering process over the multiple detections be performed as a more representative count of detections and false alarms. The false alarm reduction stage and the classification stage in Figure 2 only operate on the clustered locations. The size of the clustering region was determined by the size of targets (in this case, clustering is 22-pixel long). The clustering used is as follows: each frame (512 by 2048 pixels) of the SAR data is processed by the front-end detection stage and the outputs which exceed a threshold in (51) are stored by magnitude; the clustering starts at the location with the maximum output and groups all the raw detections within a representative range of target sizes; next, all those grouped locations are merged at a single representative location based upon the weighted sum of the outputs in the grouping. The number of false alarms is computed from the clustered detections.

The two-parameter CFAR detector yielded 4,455 false alarms over the 127 frames of the mission 90 pass 5 SAR data set when the detection threshold was set at 100% target detection (all 345 targets). Figure 31 displays some of detection and clustering results by the two-parameter CFAR detector. The detection points (false positives) in the centers of the subimages (image chips having 85 by 85 sizes) mostly exhibit high contrasts relative to their surroundings because the two-parameter CFAR detector depends on relative intensity information.

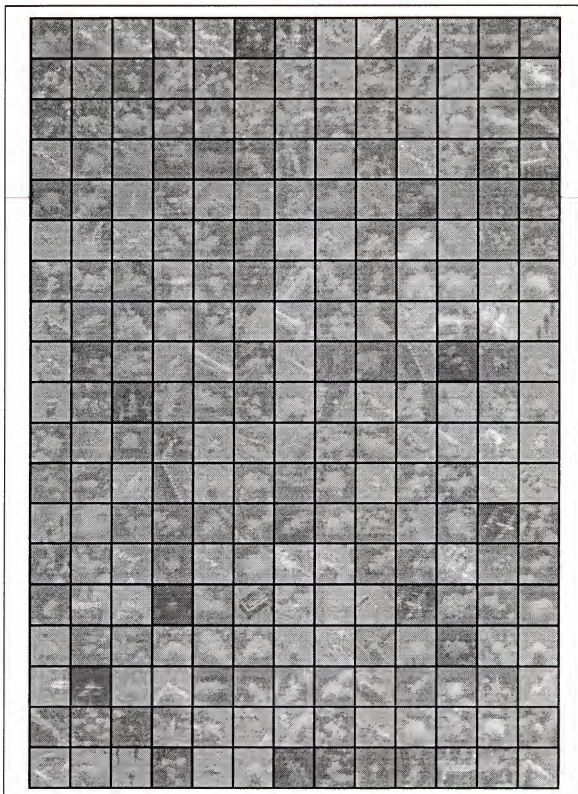


Figure 31 Some examples of non-target image chips triggered by the two-parameter CFAR detector.

6.2.2 γ CFAR Processing

6.2.2.1 Optimal Parameter Search

From the false alarms triggered by the two-parameter CFAR detector for 100% target detection, 550 false alarms as non-target image chips were randomly selected and integrated with the set of 345 target image chips to create the data base for finding optimal parameters (μ_1 and μ_{15}) of the γ CFAR detector. After the clustering process, each of the targets and non-target objects from detections was centered in the image chips by using its highest intensity pixel. Some of non-target image chips are shown in Figure 31.

The parameter space of the μ_1 and μ_{15} was incrementally scanned between 0.0304 and 4.6052. This range was converted from the range (0 to 1) in which gamma kernels in discrete time are stable: first, the range (0 to 1) was equally scanned in 33 steps for stable gamma kernels in discrete time; next, it was converted to the range (0.03 to 4.6052) by the following relationship $\mu_c = -\ln(1 - \mu_d)$ where μ_d are values of the scanned range (0 to 1) in 33 steps and μ_c the converted values for continuous time. Kernels with 33 different μ values were computed which correspond to memory depth of about 0.5 to 33 pixels for the g_1 kernel, and 8 to 495 pixels for g_{15} kernel.

The false alarms were computed in the follow way: First, a target output of the γ CFAR detector (set to detect all the targets) is found and set to be a threshold for each set of μ_1 and μ_{15} and then, with this threshold, we computed the corresponding false alarm rate in the parameter space. The false alarm surface in the μ_1 and μ_{15} space is shown in Figure 31a.

After searching the 2-D false alarm surface, the minimum false alarm rate was found with $\mu_1 = 1.0788$ (index 22, memory depth of 1 pixel) and $\mu_{15} = 0.5978$ (index 15, memory depth of 25 pixels). This combination constitutes a γ CFAR stencil which can be thought of as a local feature extractor for the best discrimination between the target and non-target classes in the training image chips. The minimum number of false alarms produced by the γ CFAR detector was 105 in the training image chips. With the optimal values

of μ_1 and μ_{15} , the guard band size was approximately 15-pixel wide, and the local area was approximated within a 10-pixel wide ring band (Figure 32b). Note that the γ CFAR detector was able to improve upon the CFAR performance which produced 550 false alarms, changing the memory depths of the 2-D gamma kernels (g_1 and g_{15}) in the stencil.

6.2.2.2 Impact of Stencil Size in False Alarm

The false alarm surface in Figure 32a illustrates the importance of the stencil for detection performance. What is important to note is the dramatic dependence of false alarm rate on the shapes of g_1 and g_{15} kernels. A small difference in the shapes of the kernels makes a big difference in the false alarm performance. The gamma kernels are continuous functions of μ 's so that the gamma kernels can be differentiated with respect to the parameter μ 's. Hence a productive way of setting the scale parameters, μ_1 and μ_{15} , is to use adaptation algorithms from adaptive filter theory.

Deciding on the shape of the stencil by the geometric characteristic of targets, as done in the two-parameter CFAR stencil, will probably give suboptimal performance. Since the 2-D gamma kernels are circularly symmetric, a single parameter of the kernels controls the kernel shapes with the kernel orders fixed. More versatile shapes may perform better, but we have to be prepared for the explosion in the degree of freedom and the inherent difficulty of setting more parameters.

It is also interesting to see the effect of changing the guard area size and the size of a target masking kernel in the CFAR stencil. In order to fairly compare the CFAR stencil with the γ CFAR stencil, circular kernels with abrupt changes in magnitude were used (Figure 33a). The size of the clutter masking kernel was set to be 10-pixel wide, to approximate the width (about 10-pixel wide) of the optimal g_{15} kernel (Figure 32b). The numbers of false alarms were computed by changing r_2 and r_1 . The minimum number of false alarms was 360 which was obtained at $r_1 = 2$ and $r_2 = 27$ (Figure 33b). This number is about 3.5 times the minimum number (105 false alarms) of false alarms by the γ CFAR

detector. This implies that the abrupt shape of the CFAR stencil may cause undesirable detection performance.

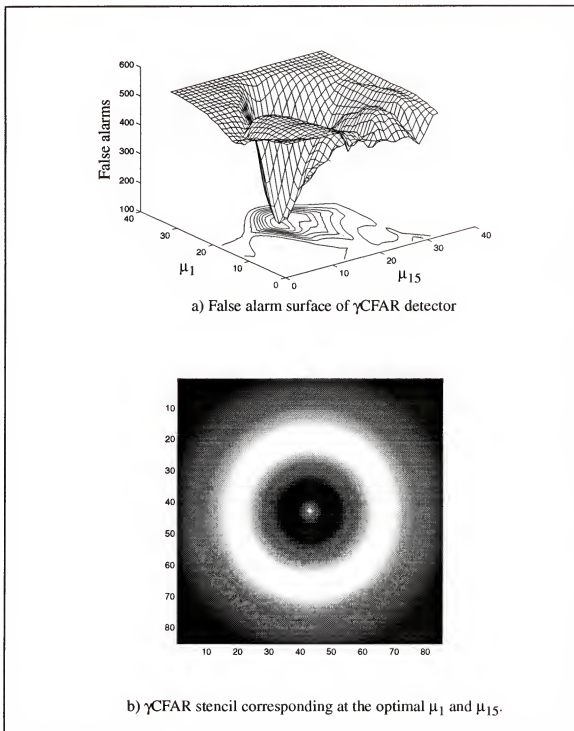


Figure 32 The false alarm surface of γ CFAR detector and the corresponding stencil at the optimal parameters.

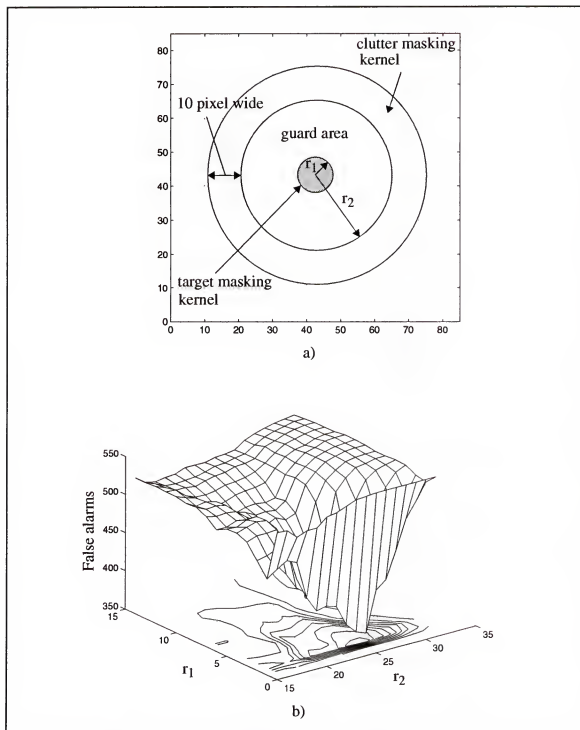


Figure 33 Impact of CFAR stencil size to false alarm: a) round-shape CFAR stencil with abrupt change in magnitude, b) false alarms versus the radius r_1 of a target masking kernel and the radius r_2 of a clutter masking kernel from the pivot point in the CFAR stencil.

6.2.2.3 Batch-running the γ CFAR Detector

With the optimal set of μ_1 and μ_{15} , the γ CFAR detector was run over the first 127 frames of the Mission 90 Pass 5 SAR data set with the same embedded targets as for the CFAR detector. The size of the γ CFAR stencil was 85 by 85 pixels as for the CFAR stencil.

In order to compute the γ CFAR output at each pixel in the image, three features (1st moment at the center pixel and 1st and 2nd moments in the local region) are required to be computed by convolving the image with two gamma kernels ($g_{1,\mu}$ and $g_{15,\mu}$). The convolution was computed in the frequency domain using FFTs to obtain better computational efficiency. The image sequence was divided into overlapping radix 2 windows (2048 by 128 pixels) and processed by an overlap and save method. After processing the entire imagery of the Mission 90 Pass 5 SAR data set by the γ CFAR detector, the minimum output value was selected to be a threshold for 100% target detection. All detection points above the threshold were clustered as for the CFAR case.

6.2.3 Performance Comparison of the Two-Parameter CFAR Detector and the γ CFAR Detector

The performances of the two-parameter CFAR detector and the γ CFAR detector are compared by ROC curves (Figure 34). For 100% detection, the γ CFAR detector yielded 760 false alarms while the CFAR detector produced 4,455 false alarms (a 1:6 ratio). Less false alarms created by the prescreener means that the computational bandwidth of the subsequent processing modules can be decreased. With a discount of 2% target outliers, the γ CFAR and the CFAR detectors yielded 239 and 510 false alarms respectively for 98% target detection.

Overall, the ROC of the γ CFAR in Figure 34 shows more robust performance than the standard two parameter CFAR test. Figure 35 shows the number of raw detections (false positives) and clustered detections per frame. The γ CFAR detector outperformed the

detection performance of the two-parameter CFAR detector in all frames, i.e. both for natural and cultural clutter. We can observe that the number of false positive change from frame to frame. It is also interesting to note that clustering merges the raw detections into smaller numbers of ROIs in each frame. The overall computation requirements by the two-parameter CFAR detector is larger for clustering due to an excessive false positives.

Figure 37 and Figure 38 display the performance of both detectors in areas having different statistical characteristics, i.e. natural clutter area from frames 7 and 8 and a cultural clutter area from frames 121 and 122 (Figure 36). In the cultural clutter region, the γ CFAR detector produced 59 false alarms the CFAR detector yielded 113 false alarms. In the natural clutter region, 4 false alarms occurred by the γ CFAR detector and 28 false alarms by the CFAR detector. Note that the objects in the boxes in Figure 36b indicate targets embedded.

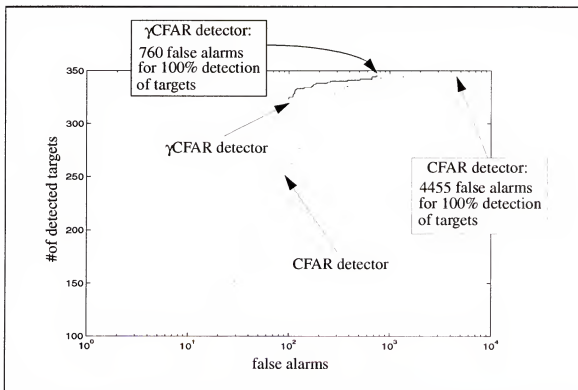


Figure 34 Global ROC measurement over 127 frames of the Mission 90 pass 5 SAR data set.

In Figure 38, multiple detections occurs on tops of the targets and brighter objects of man-made clutter because these are typically high contrast relative to their background clutter. The two-parameter CFAR detector produced considerable amounts of detections along the roof edges of the houses toward the SAR (that is, upwards the figure). Many detections also occur on tops of the trees in natural clutter because the relative contrasts between trees and their surrounding radar shadows are large which causes the CFAR detections.

6.2.4 Conclusion

Unlike the two-parameter CFAR detector which uses an a priori defined stencil, the γ CFAR detector utilizes the family of 2-D gamma kernels where the free parameter μ changes the shape and scale of the γ CFAR stencil. This work shows that the performance of the standard two parameter CFAR detector can be improved if the size of the stencil is optimized. It is true that the shape of the γ CFAR is graded and also different from that of the CFAR stencil, but one of the most important factors for the performance improvement is the stencil size. The optimal value of μ was found through exhaustive search just to quantify what is the best possible performance. In an adaptive signal processing framework, we can find the best value of μ through training, either by minimizing false alarms or by minimizing an output error measure. This has been done already for the 1-D case [64], and the methodology can be extended to the 2-D case.

The big challenge is to find an on-line performance criterion that will be able to decide what is the best scale. Further research should also look into the use of nonsymmetric kernels. The γ CFAR can be interpreted as an estimator of the local intensity statistics. In this perspective the use of the full kernel, instead of using only g_1 and g_{15} motivated by the CFAR stencil, should improve the performance of the detector.

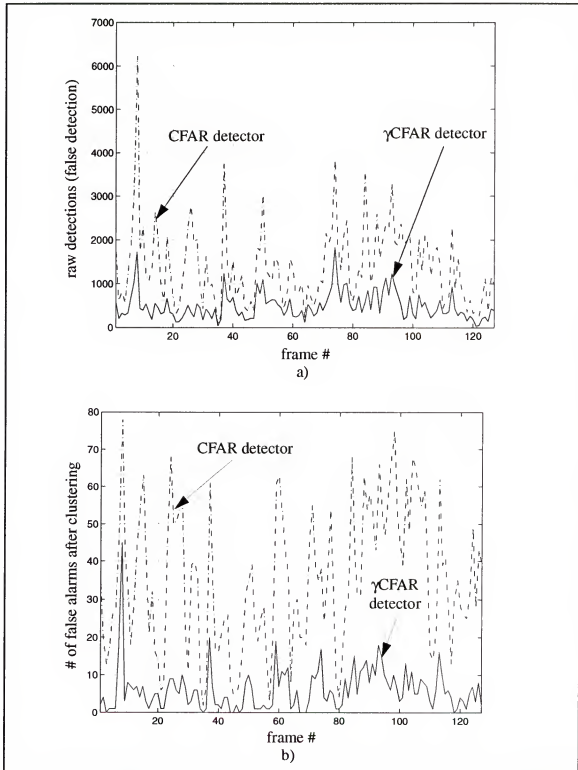


Figure 35 False alarms vs. frame. a) The number of raw detections (false detections) was counted before clustering, and b) the number of false alarms was counted to detect 345 targets (testing target set) after clustering through 127 frames of the Mission 90 Pass 5 SAR data set.

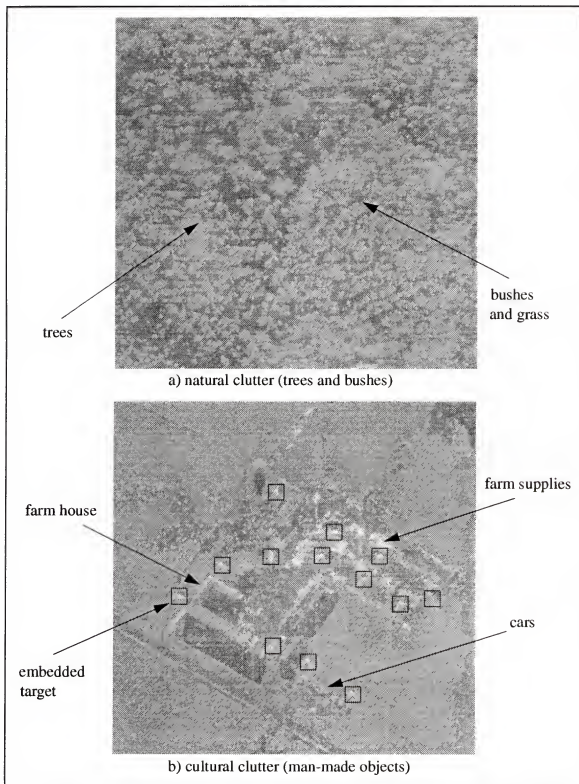
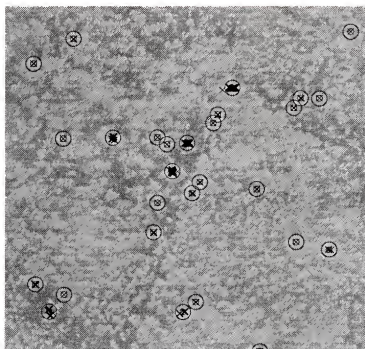


Figure 36 A natural clutter region (a) from frames 121 and 122 and cultural clutter region (b) from frames 7 and 8 in the mission 90 pass 5 data set. In the cultural clutter, 13 targets were embedded and marked by rectangular boxes.



a) CFAR detector

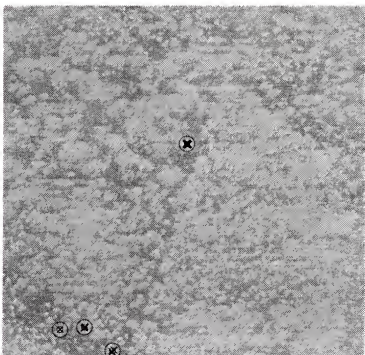
b) γ CFAR detector

Figure 37 The performance comparison of CFAR and γ CFAR detector. The CFAR detector yielded 28 false alarms while the γ CFAR detector triggered only 4 false alarms.

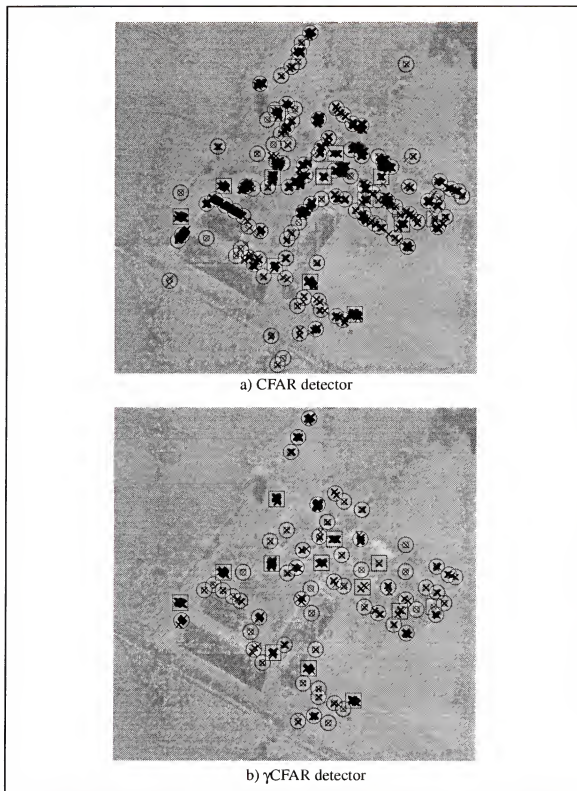


Figure 38 The performance comparison of CFAR and γ CFAR detector. The CFAR detector yielded 113 false alarms while the γ CFAR detector triggered only 59 false alarms.

6.3 False Alarm Reduction

6.3.1 CFAR/QGD

6.3.1.1 Training QGD

6.3.1.1.1 Training Data Preparation

In order to train the QGD, 275 embedded target image chips of a training set were used for the target class and 550 clutter image chips were randomly selected from the 4455 false alarms caused by the two-parameter CFAR detector. Before the target and clutter image chips were collected, the clustering was performed and multiple detections within a target size (23 pixel long) were clustered into a centroid.

6.3.1.1.2 Optimal Weights by the Closed Form Solution

The least squares method is utilized to optimally compute the weights of the QGD in the training set. The weights are found such that the power of differences between the system outputs and the corresponding desired responses are minimized (L_2 norm).

Since the size of the training set (550 clutter and 275 target image chips) is greater than the number of weights (8 weights), the method solves an overdetermined system of linear equations by (85), and is a parametric least squares problem because the least squares solution is also dependent of the parameters, μ_1 and μ_{15} . The parameters and weight vector of the QGD can be obtained through an exhaustive search in the parameter space. Given each combination of μ_1 and μ_{15} , the corresponding least squares solution is computed. Optimal parameters and weight vectors are determined in which the number of false alarms is minimum in the training set.

Figure 39a shows the false alarm surface in the parameter space μ . The optimal indices of the parameters, μ_1 and μ_{15} , are 8 and 16 which correspond to 0.274 and 0.654 respectively. Note that the optimal values of μ_1 are much smaller than those of μ_1 in the γ CFAR detector, which means that the shape of the kernel g_1 of the QGD is much broader.

Since, in the training set of the QGD, the clustering points are the centers of mass of detections, the brightest pixels are not always placed at the centers of image chips. The g_1 kernel of the QGD stretches out to capture the bright pixel intensities which usually help discriminate targets from clutter. So g_1 ends up being a broader shape of kernel compared to the that in the γ CFAR stencil.

Another distinguishing feature of the QGD false alarm surface is that it shows the small numbers of false alarms at several different location in the parameter surface (local minima in the false alarms). This is because, in addition to the parameters (μ 's), the QGD has 8 weights which allow for more discriminating power than the γ CFAR detector. That is, the optimal weights counterbalance improper choices of the parameters.

The SSEs of the QGD were computed in the parameter space as for the false alarms and the SSE surface is shown in Figure 39b. Compared to the false alarm surface, the SSE surface is much smoother and the optimal indices of μ_1 and μ_{15} yielding the least SSE are 4 and 16 which correspond 0.128 and 0.654. The optimal μ_{15} in the false alarm surface is the same as the optimal μ_{15} in the SSE surface but the least SSE is found at a smaller value of the μ_1 , compared to the minimum false alarm.

However, the two optimal points in both the SSE and the false alarm surfaces are close to each other in the parameter space. This implies that, as for the case of the LS method with an exhaustive search of the parameter space (w_1, \dots, w_8), the QGD may be able to obtain a comparable false alarm performance from an gradient descent method by adaptively computing the free parameters ($\mu_1, \mu_{15}, w_1, \dots, w_8$) when the SSE approaches nearby the global minimum point in the 10-dimensional parameter space ($\mu_1, \mu_{15}, w_1, \dots, w_8$) during the training.

Table 1 Detection performance of QGD in the training set.

Quadratic Gamma Detector (QGD)						
Target detection rates	100%	99%	98%	95%	92%	SSE
# of false alarms	14	14	8	4	3	0.0273

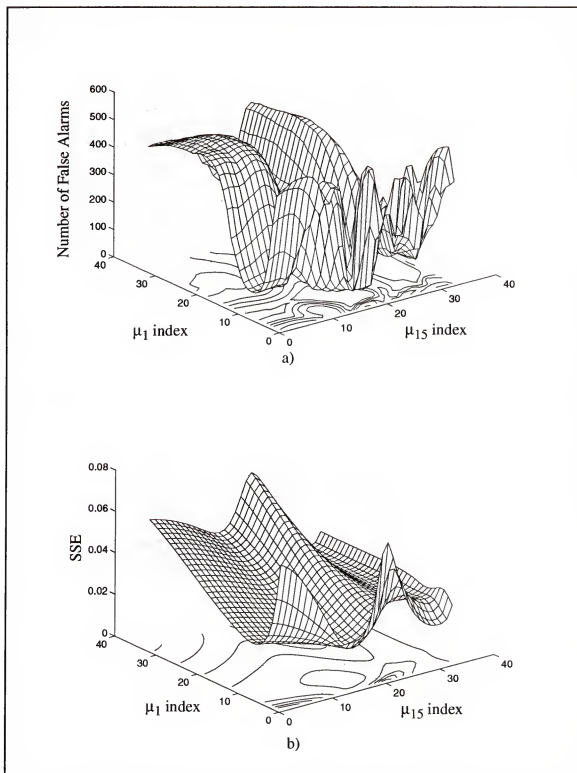


Figure 39 Performance surfaces of the QGD in parameter space. a) False alarm surface, b) SSE surface with optimal weights.

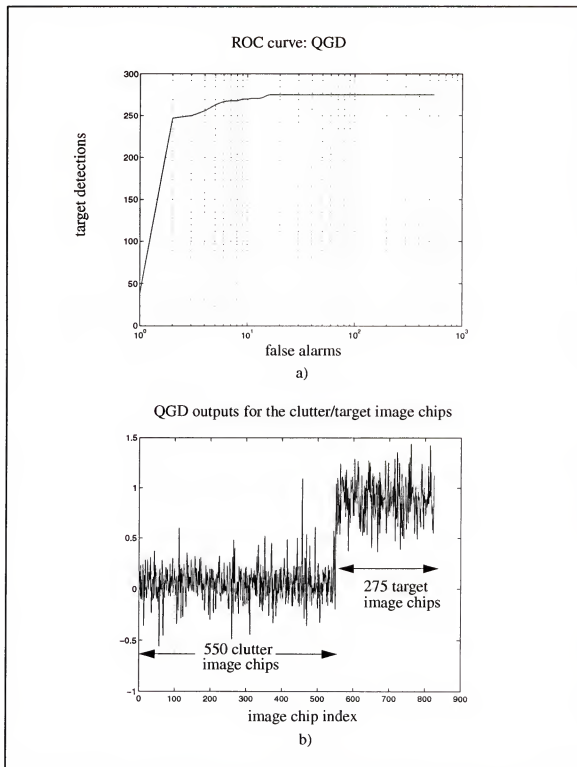


Figure 40 The discriminating performance of the QGD in the training set. a) ROC curves, b) QGD outputs for the inputs of clutter and target image chips after training.

In order to detect all targets (275 targets) in the training set, the QGD yielded 14 false alarms at 100% and 99% detection rates, and for 98%, 95% and 92% target detections, 8, 4, and 3 false alarms occurred respectively in Table 1. As shown in Figure 40b, clutter and target outputs were well separated.

6.3.1.1.3 Testing the QGD

Given a set of parameters, the QGD training seeks optimal weights based on minimizing the sum of squared errors between the outputs and their corresponding desired responses. The optimal set of parameters (μ_1 and μ_{15}) are determined which leads to the minimum number of false alarms in the training set. With the optimal set of parameters and optimal weights, the QGD was tested on different sets of clutter and target image chip embeddings. Table 2 shows the performance of the QGD in the testing phase based on SSE and false alarms given detection rates.

The QGD exhibits a discrimination power of reducing 3905 false alarms to 385, 118, 97, 53 and 42 false alarms for 100%, 99%, 98%, 95%, and 92% detection rates respectively. For all clutter image chips (4455 false alarms caused by the two-parameter CFAR detector), the false alarms were reduced to 422, 132, 109, 57, and 44 at 100%, 99%, 98%, 95%, and 92% detection rates respectively by the QGD. A discrimination power of about 1:10 ratio (422/4455) was obtained by the QGD at 100% target detections in the false alarm reduction stage in the ATD/R system (Figure 2).

6.3.1.2 Training the QGD in an Iterative Manner

An optimal weight set of the QGD was computed based on the LS given μ_1 and μ_{15} in the previous section. The optimal parameter set (μ_1 and μ_{15}) was selected based on the minimum false alarm for 100% detection after an exhaustive search of the 2-D parameter space.

Here the weights and the parameters (μ_1 and μ_{15}) were adaptively computed in an

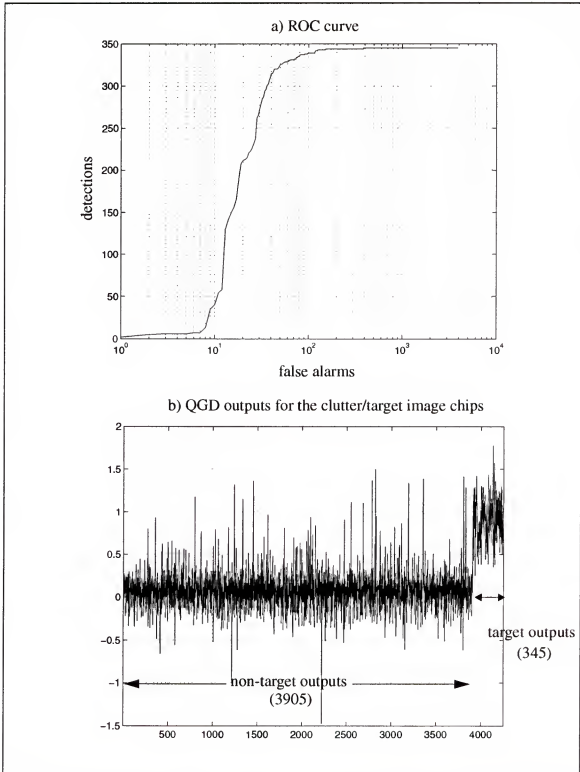


Figure 41 Testing results of the QGD for testing set #1.1. a) ROC curve, b) outputs of the QGD.

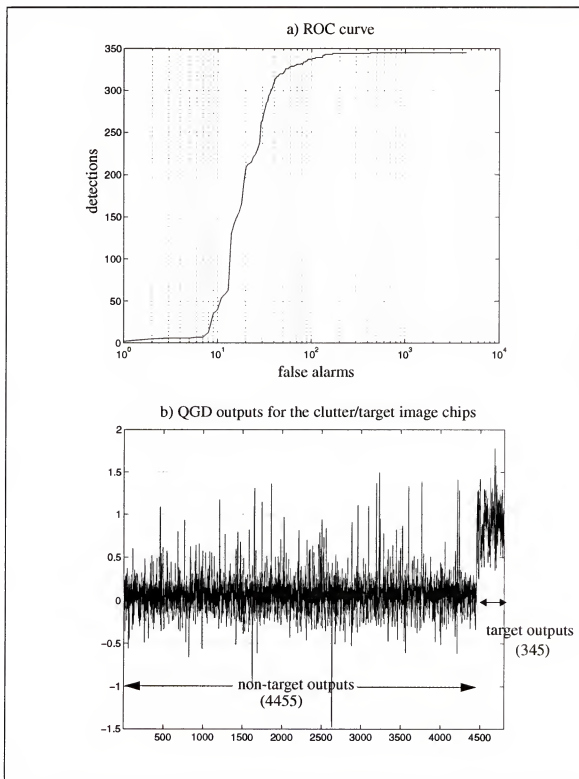


Figure 42 Testing results of the QGD for testing set #2. a) ROC curves, b) outputs of the QGD.

Table 2 Detection performance of QGD in testing.

Quadratic Gamma Detector (QGD)						
Target detection rates	100%	99%	98%	95%	92%	SSE
No. of false alarms in testing set #1.1*/No. of false alarms in testing set #2*	385/ 422	118/ 132	97/ 109	53/ 57	42/ 44	0.0202/ 0.0192

*Note that the testing set #1.1 contains 3905 clutter image chips (the image chips except 550 training clutter image chips out of 4455 false alarms by the two-parameter CFAR detector) and 345 target image chips embedded over 127 frames (about 7 km^2) of the mission 90 pass 5 data set for testing purpose. The testing set #2 includes all the clutter image chips (4455) and 345 target image chips embedded for testing.

iterative manner by a gradient descent method during the training. At each iteration, the learning and momentum rates were changed to accelerate the convergence speed of the weight and parameter variables, that is, when the current error is smaller than the error at the previous iteration the learning and momentum rates were increased by small amounts and otherwise decreased to 70% of the values of the learning and momentum rates at the previous iteration. In the QGD feature set, the intensity mean values is much smaller than the intensity squared mean. This causes a wide spread in optimal weight values. So the weight and μ adaptation were performed based on a whitened feature set at each iteration. The training was stopped when the change in the number of false alarms for 100% detection was less than two false alarms for 3500 iterations.

Figure 43a and Figure 43b show the learning curve and false alarm curve of the QGD. The number of false alarms for 100% detection decreases as the error decreases during the training.

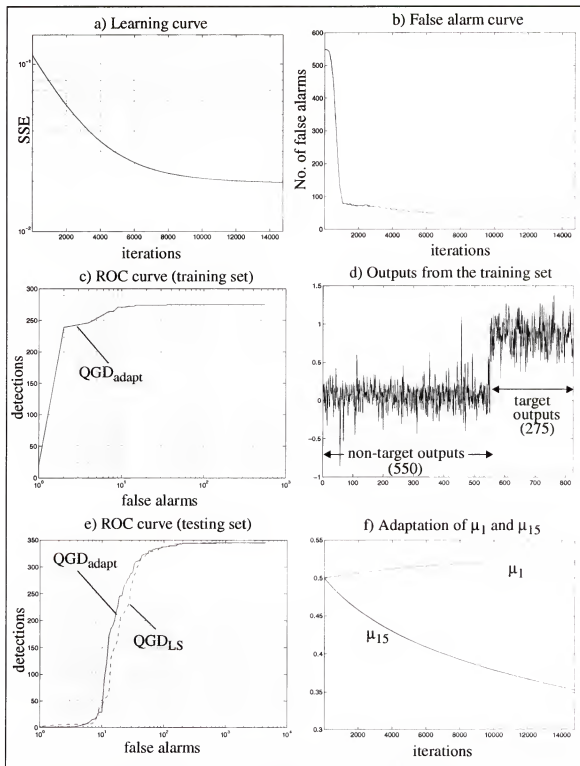


Figure 43 a) Learning curve, b) False alarm curve for 100% detection, c) ROC curve after training the QGD, d) Outputs of the QGD from the training set after training the QGD, e) ROC curve from the testing set, and f) Adaptation of the parameters (μ_1 and μ_{15}).

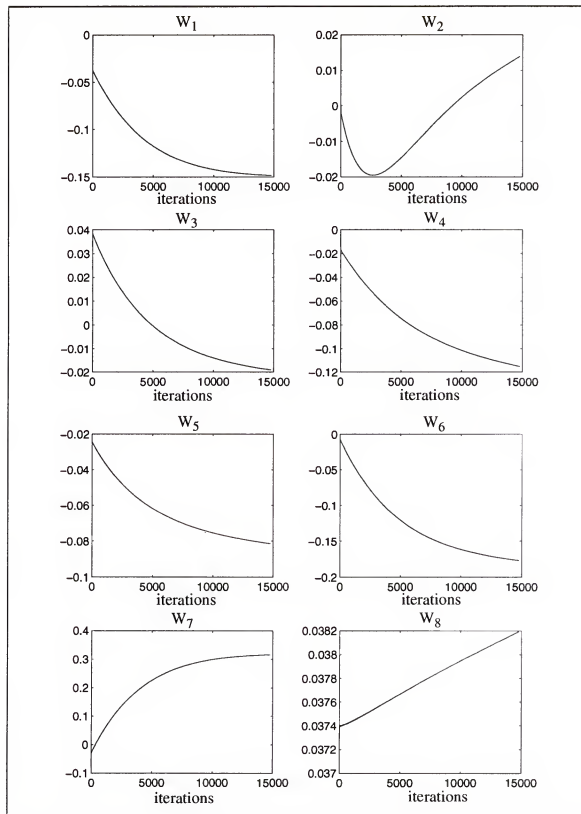


Figure 44 Adaptation of weights

In the LS solution, the optimal set of parameters were chosen based on the number of minimum false alarms. In this iterative method, the parameters were adaptively computed in the sense of minimizing the SSE. In Figure 43f and Figure 44, the parameter (μ_{15}) and the weights (W_2 and W_8) didn't seem to converge to certain values. Since the training was stopped based on the change in the false alarms for 100% target detection, the outputs of the QGD after the training was stopped were well separated (Figure 43d). Table 3 demonstrates the detection performance of the QGD trained in the iterative manner.

Table 3 Detection performance of the adaptively trained QGD in the training set.

Quadratic Gamma Detector (QGD)						
Target detection rates	100%	99%	98%	95%	92%	SSE
# of false alarms	35	14	8	6	4	0.0196

The detection performance of the iteratively trained QGD is inferior to the QGD trained on the LS at probabilities of detection (100% ~ 92%) in the training set.

Table 4 Detection performance of the adaptively trained QGD in the testing set.

Quadratic Gamma Detector (QGD)						
Target detection rates	100%	99%	98%	95%	92%	SSE
# of false alarms	598/ 695	142/ 162	106/ 122	48/ 55	36/ 42	0.0281/ 0.0273

In the testing set, as in the case of the training set, the iteratively trained QGD yielded more false alarms at high probabilities (100% ~ 98%) of detection (Table 4). Below about 95%, the detection rate is higher than that of the QGD trained on the LS.

6.3.1.3 Independent Testing Results for the QGD

A more complete study of the QGD was independently conducted at MIT Lincoln Laboratory using a large SAR database of real military targets and ground clutter. The objective of this study was to evaluate the performance of the QGD in aiding the detection stage of the SAR ATD/R system. The data for this first test case consisted of high resolution ($1ft \times 1ft$) fully polarimetric SAR imagery preprocessed using the PWF.

The QGD was trained on two target types using spotlight target data and also man-made discretely from stripmap clutter data. A total of 135 target image chips were chosen for training; these were 5 degrees apart in aspect angle (i.e., 5, 10, 15 degrees, etc.). The clutter data used for training consisted of 100 typical man-made discretely. Evaluation of this test case was performed using spotlight target and stripmap clutter data. As in the training stage, spotlight data of two targets that were 5 degrees apart in aspect angle (i.e., 3, 8, 13, 18 degrees, etc.) were used for testing. The test clutter data consisted of 4727 stripmap clutter image chips extracted from a total of 56 km^2 in area. Thus, the test data set for this experiment was composed of 139 target image chips and 4727 clutter image chips.

The QGD was evaluated by running the data through the two-parameter CFAR detector first (i.e. only over the image chips that triggered by the two-parameter CFAR detector). Then, the ROC curves were obtained by computing the cumulative number of false alarms out of each detector. At a probability of detection of 1.0 ($P_d = 1.0$), the CFAR algorithm detected 139 targets and had 2499 false alarms, whereas the QGD, while also detecting 139 targets, reduced the above-mentioned false alarm number to 715 (Figure 45).

The second test case study used single channel (HH) stripmap imagery with a resolution of $1m \times 1m$. The training set for the QGD consisted of 52 target image chips and 150 clutter image chips that represented two types of targets and man made clutter. The evaluation of the QGD was performed using 75 target image chips and 44599 clutter image chips, which triggered the two parameter CFAR detector when analyzing a 231 km^2 of area.

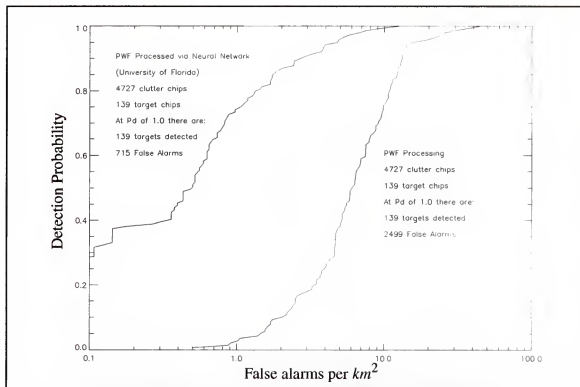


Figure 45 Discriminant performance of the QGD versus CFAR.

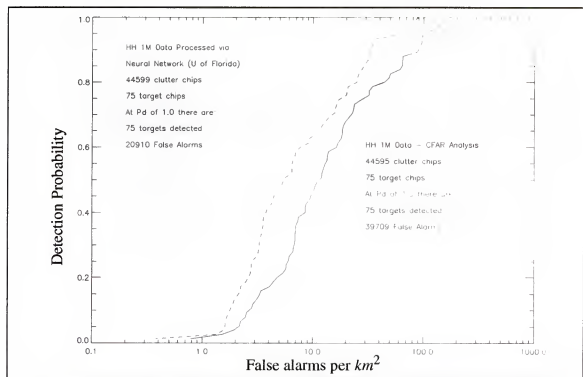


Figure 46 Comparison of the two parameter CFAR and the QGD for 1 polarization, 1 meter data.

Figure 46 shows the results. At $P_d = 100\%$, the two parameter CFAR detector had 39,709 false alarms, while the QGD also detected all 75 targets and had only 19,037 false alarms.

These two experiments constitute a comprehensive test of the performance of the QGD. As we expected the generalization capability of the QGD is very good, which should not be surprising, since the QGD has only 8 free parameters. Consequently, in practice the QGD can be trained for robust performance. The better discriminant characteristics of the QGD can be due to the better estimates for the mean and standard deviation for targets and clutter, or due to the larger feature space.

6.3.1.4 Conclusion

The appeal of the QGD is that the scale of the regions where the statistics are estimated can be adapted during training with the error at the output of the detector. The implementation that we chose uses only a subset of the gamma kernels (g_1 and g_{15}). It was derived by analogy with the two parameter CFAR detector to enable a straight forward comparison with this widely used algorithm. In the QGD, the estimates of the local statistics are obtained by convolution with the g_1 (cell under test) and g_{15} kernels (local neighborhood). The QGD is superior to the conventional two parameter CFAR in terms of detection performances. The Lincoln Laboratory testing also shows that the QGD can improve the false alarm rate of the two parameter CFAR detector without affecting the probability of detection.

6.3.2 CFAR/NL-QGDs

6.3.2.1 L_2 Based Training and Testing of the NL-QGDs

In order to compare the performance of the QGD and NL-QGD based on the same feature values, the same optimal parameters for the QGD, μ_1 and μ_{15} , were used for the NL-QGD. Since the QGD features are highly correlated one another it is desirable to whiten the training data set for the adaptation of weights in the NL-QGD training to

achieve fast convergence speed. The following matrix indicates the correlation coefficient matrix of the QGD training set. As seen in the matrix, the first column vector shows high correlation among the values of the 1st, 3rd 5th and 7th rows $((g_1 \bullet X), (g_1 \bullet X^2), (g_1 \bullet X)^2, (g_1 \bullet X) (g_{15} \bullet X))$, the second column exhibits high correlation among the values of the 2nd, 4th, 6th and 7th rows $((g_{15} \bullet X), (g_{15} \bullet X^2), (g_{15} \bullet X)^2, (g_1 \bullet X) (g_{15} \bullet X))$ and so on.

1.0000	0.7733	0.9946	0.7617	0.9971	0.7677	0.9558
0.7733	1.0000	0.7287	0.9959	0.7478	0.9961	0.9192
0.9946	0.7287	1.0000	0.7205	0.9991	0.7276	0.9372
0.7617	0.9959	0.7205	1.0000	0.7401	0.9995	0.9163
0.9971	0.7478	0.9991	0.7401	1.0000	0.7468	0.9472
0.7677	0.9961	0.7276	0.9995	0.7468	1.0000	0.9204
0.9558	0.9192	0.9372	0.9163	0.9472	0.9204	1.0000

By using the whitening transformation [27], the training data set was orthogonalized and the resulting correlation coefficient matrix becomes an identity matrix as follows;

1.0000	0.0000	0.0000	0.0000	-0.0000	0.0000	-0.0000
0.0000	1.0000	0.0000	-0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	1.0000	0.0000	0.0000	-0.0000	-0.0000
0.0000	-0.0000	0.0000	1.0000	-0.0000	-0.0000	-0.0000
-0.0000	0.0000	0.0000	-0.0000	1.0000	0.0000	-0.0000
0.0000	0.0000	-0.0000	-0.0000	0.0000	1.0000	-0.0000
-0.0000	0.0000	-0.0000	-0.0000	-0.0000	-0.0000	1.0000

Backpropagation with a momentum term was used to train the NL-QGD weights with the whitened training data set. The cross validation set was also whitened by using the estimate of the covariance matrix of the training data set. Training was stopped when the error in the cross-validation set started to increase.

The learning and false alarm curves of the NL-QGD with 5 hidden nodes were plotted in Figure 47. With the L_2 normed training, it is observed in Figure 47b that the number of false alarms for 100% target detection in the cross validation and training sets decreased as the SSE decreased for about first 200 training epochs. However, the false alarms increase

between about the 200th and 1000th epoch and show saturation after about the 1000th epoch. This implies that the outliers of the target class produce very low output values after the training and the L_2 norm may not be a good criterion for training a detector. So the target outliers should be prevented from producing very low output values.

Figure 48 shows the ROC plots and outputs of the QGD and NL-QGD after the training. Figure 49 and Figure 50 plot the ROC curves and the outputs of the NL-QGD751 in the testing sets.

The performance of the QGD and NL-QGD are compared for different network sizes and the SSE and the number of false alarms are presented for both systems in Table 5 and Table 6. In the training set, the ROC's of the QGD and the NL-QGD are comparable. In the testing sets, the NL-QGD outperforms that of the QGD with respect to detection probability at most detection rates. The NL-QGD is able to provide a smaller final SSE than the QGD. However, the performance of a detector is not measured in terms of SSE but number of false alarms for a given detection accuracy. Hence, the ROC curve of the two systems must be compared. The comparisons were restricted to one hidden layer networks, and the hidden layer size was increased from 3 to 7 nodes. The NL-QGD did not perform as well as the QGD at the 100% detection rate. The detector output of the NL-QGD tended to provide large misclassifications which affected the selection of the threshold. However, at 99%, 98% and 95% detection probability the NL-QGD outperformed the QGD. Note that changing the number of hidden nodes from 3 to 7 did not seem to affect the performance of the NL-QGD much.

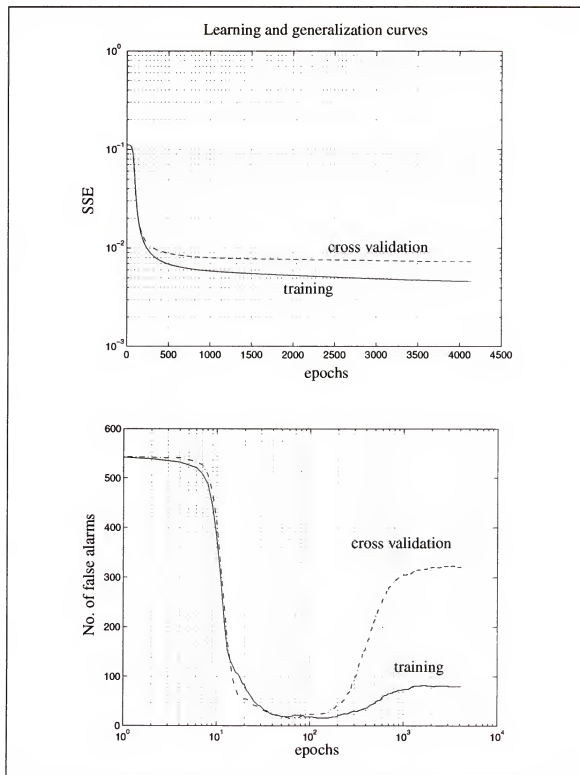


Figure 47 Learning and False alarm curves of the NL-QGD751 trained based on L_2 norm.

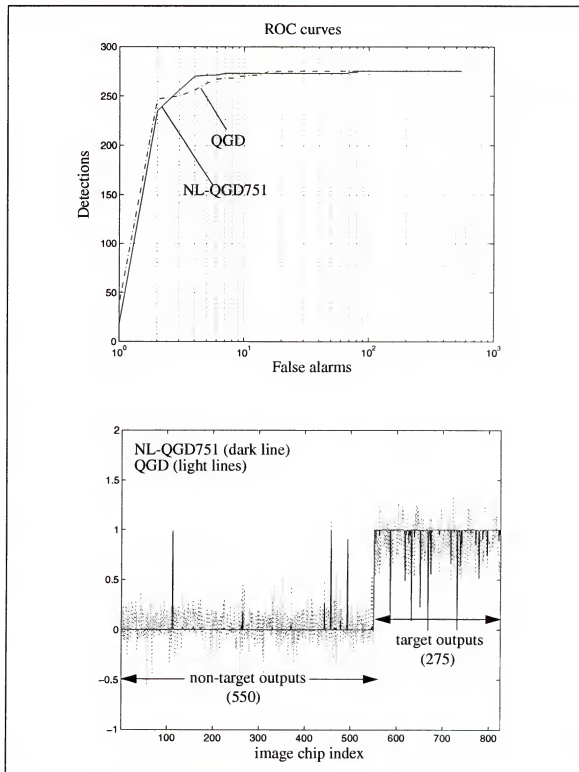


Figure 48 Training results of the L_2 normed NL-QGD751. a) ROC curves, b) outputs of the NL-QGD751 for the training set.

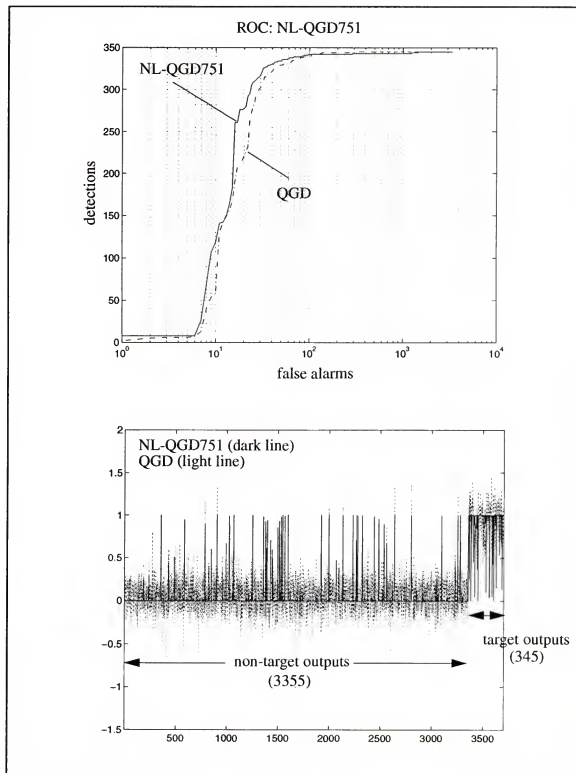


Figure 49 Testing results of the L_2 normed NL-QGD751 for testing set #1.1. a) ROC curves, b) outputs of the NL-QGD751.

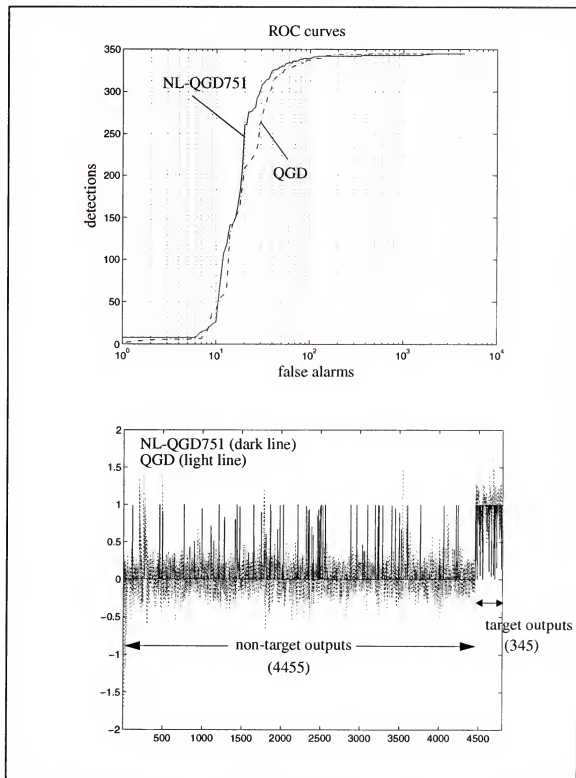


Figure 50 Testing results of the L_2 normed NL-QGD751 for testing set #2. a) ROC curves, b) outputs of the NL-QGD751.

Table 5 Detection performance of NL-QGD in training set.

Network Topologies*	Detection Rates					SSE	Stop iterations
	100%	99%	98%	95%	92%		
NL-QGD731	80	6	3	3	2	0.005039	4156
NL-QGD751	80	6	3	3	2	0.005038	4139
NL-QGD771	80	6	3	3	2	0.005048	4117

* The NL-QGD731 indicates that the network has 7 input nodes, 3 nodes in the hidden layer, and 1 output node. For NL-QGD training, the learning rate (η) and momentum rate (α) were 0.1 and 0.02 respectively for all the networks.

Table 6 Detection performance of NL-QGD in testing.

Network Topologies	Detection Rates (No. of false alarms in testing set #1.2* / No. of false alarms in testing set #2)					SSE
	100%	99%	98%	95%	92%	
QGD _{LS}	340/ 422	105/ 132	85/ 109	45/ 57	35/ 44	0.0204/ 0.0192
QGD _{dapt}	514/ 695	115/ 162	86/ 122	41/ 55	32/ 42	0.0292/ 0.0273
NL-QGD731	1456/ 1892	96/ 120	67/ 83	35/ 46	30/ 37	0.006263/ 0.005848
NL-QGD751	1453/ 1889	96/ 120	67/ 83	35/ 46	30/ 37	0.006261/ 0.005846
NL-QGD771	1445/ 1880	97/ 120	67/ 83	35/ 46	30/ 37	0.006262/ 0.005847

*Note that the testing set #1.2 contains 3355 clutter image chips (the image chips except 550 training and 550 cross validation clutter image chips out of 4455 false alarms by the two-parameter CFAR detector) and 345 target image chips embedded over 127 frames (about 7 km²) of the Mission 90 Pass 5 SAR data set for testing purpose.

The NL-QGD performed better than the QGD in most of detection probabilities. However, at 100% detection its performance is inferior to the QGD. We can observe that the NL-QGD did a much better job in terms of SSE by analyzing the detection outputs (Figure 48b). When it gave the wrong output, the error was much larger than the QGD's. We can expect this from the nonlinear nature of the detector. In terms of number of false alarms, this behavior affects the performance because the threshold for 100% detection has to be set based on the smallest value obtained from the target image chips.

6.3.2.2 Training and Testing of the NL-QGDs Without Non-Target Outliers

In the L_2 normed training, it was observed that the NL-QGDs produced very low output values for the target outliers and high values for the non-target outliers. By removing the non-target outliers during the training, the decision boundaries of the NL-QGDs move toward the target outliers so that the output values of the NL-QGDs for the non-target outliers increase. This leads to a threshold for a high probability of target detection to increase.

The training procedure for the NL-QGDs are as follows: first, the NL-QGDs are trained based on L_2 norm and the number of false alarms is computed at each iteration during the training, second, when the number of false alarms in the cross validation set starts increasing, the non-target sample which produced the maximum output is removed from the training set; third, this training procedure proceeds until the number of false alarms does not change for 250 iterations.

Figure 51 shows the ROC curve and the outputs of the NL-QGDs after the trained was stopped. As expected, the output values from the target outliers were significantly increased with high output values from the non-target outliers. This is because the decision boundary moved toward the target outliers so that the distances between the non-target outliers and the decision boundary became larger.

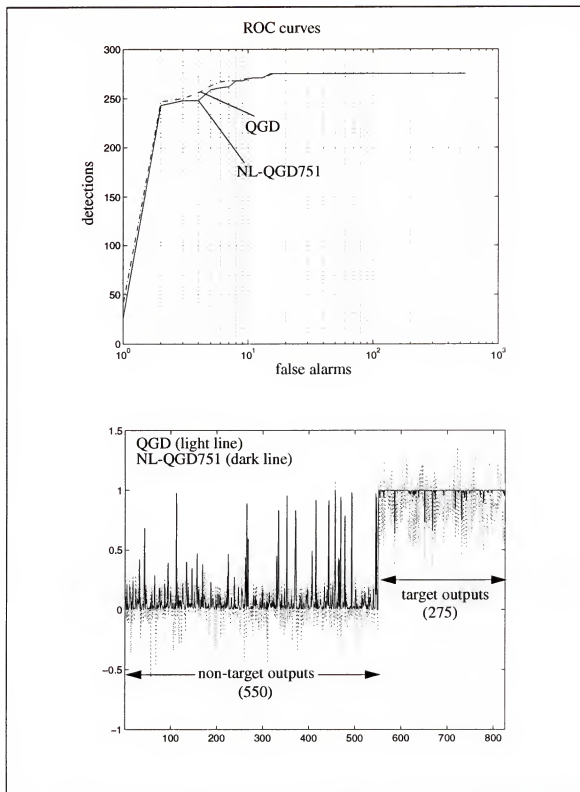


Figure 51 Training results of the NL-QGD751 trained without non-target outliers. a) ROC, b) outputs of the NL-QGD751 for the training set.

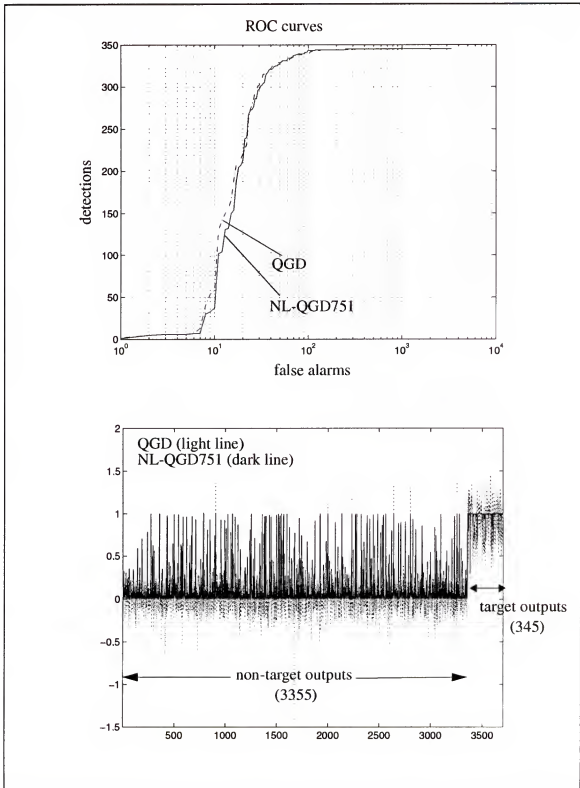


Figure 52 Testing results of the NL-QGD751 trained without non-target outliers for the testing set #1.2. a) ROC curves, b) outputs of the NL-QGD751.

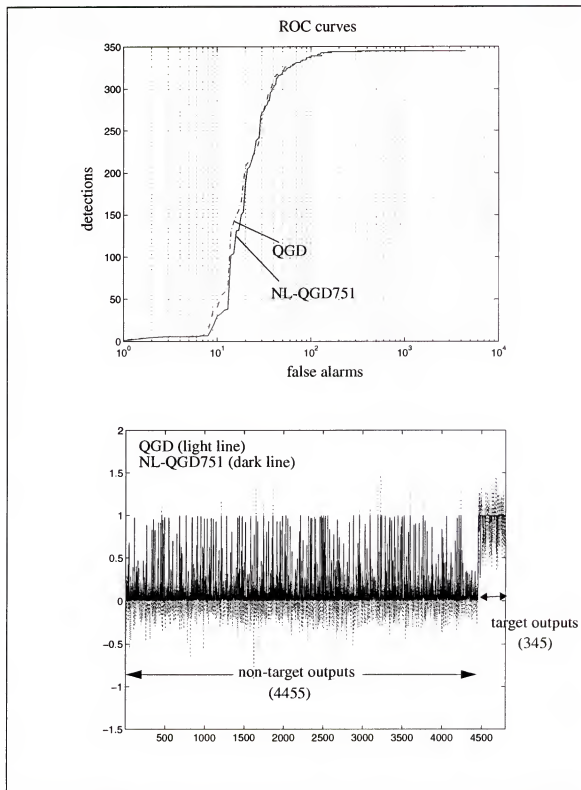


Figure 53 Testing results of the NL-QGD751 trained without non-target outliers for the testing set #2. a) ROC curves, b) Outputs of the NL-QGD751.

Table 7 summarizes the training results of the NL-QGDs trained on L_2 norm with removal of non-target outliers during training for different network sizes (3, 5 and 7 nodes in the hidden layer). During training, the numbers of non-target outliers from the training set were 8, 9, and 6 respectively. Note that the stop iterations for all network sizes are extremely small compared to the case without removal of non-target outliers. Figure 52 and Figure 53 show the ROC curves and the outputs of the QGD751 from the two testing sets. The minimum target output value is much higher than that from the NL-QGD751 trained on L_2 without removal of non-target outliers.

Table 7 Detection performance of the NL-QGDs trained without non-target outliers* (training set)

Network Topologies	Detection Rates					SSE	# of target outliers removed	Stop iterations
	100 %	99%	98%	95%	92%			
NL-QGD731	13	13	8	4	3	0.00648	8	567
NL-QGD751	14	13	9	6	4	0.00618	9	342
NL-QGD771	14	12	9	5	5	0.00630	6	426

* For NL-QGD training, the learning rate (η) and momentum rate (α) were 0.196 and 0.02 respectively for all the networks.

In the testing sets, some of the non-target outputs are very large due to the boundary shift towards the target outliers. The detection performance for the NL-QGDs in the testing sets are summarized in Table 8. The number of false alarms at 100% detection rate has significantly reduced compared to the L_2 case without removal of non-target outliers for all network sizes (3, 5, 7 nodes). The detection performance at 99% detection rate is slightly better than the L_2 case without removal of non-target outliers. Below 98% detection rate, the performance became worse because the training with removal of non-target outliers led the NL-QGDs to lose their generalization capability.

Table 8 Detection performance of the NL-QGD trained without non-target outliers. (testing sets)

Network Topologies	Detection Rates (No. of false alarms in testing set #1.2/No. of false alarms in testing set #2)					SSE
	100%	99%	98%	95%	92%	
NL-QGD731	293/ 365	103/ 130	74/ 96	45/ 58	35/ 45	0.01047/ 0.01030
NL-QGD751	250/ 315	97/ 125	77/ 101	49/ 63	37/ 49	0.01768/ 0.01741
NL-QGD771	246/ 311	98/ 126	74/ 97	48/ 61	37/ 48	0.01624/ 0.01596

6.3.2.3 L_p Based Training and Testing of the NL-QGD

As observed in the NL-QGD training based on L_2 , some of target outputs of the NL-QGDs are very low. Since the smallest value of the NL-QGD output is set to be thresholded for 100% detection, low target outputs indeed cause considerable amounts of false detections at high detection probability.

A way of alleviating large deviations of the NL-QGD outputs from their desired values was to propose to using a larger norm ($p > 2$) than the L_2 norm. By choosing $p = 8$ as a large norm, a BP algorithm with momentum was used to train the NL-QGDs. The L_8 norm is an appropriately large norm in penalizing large errors. The same set for training and cross validation was used to training the NL-QGDs. The training was stopped when the cost in the cross validation set started increasing.

Figure 54 shows the learning and generalization curves, and false alarm curves of the NL-QGD751 in the training and the cross validation sets. As opposed to the case in the L_2 norm based training, the false alarm curves decreased as the training epochs increased

with a period of false alarm increment between about the 200 th and 1000 th epoch. The false alarm decrement is somewhat in agreement with a minimizing process of the costing during training.

Table 7 illustrates the detection performance of the NL-QGD with 3, 5, and 7 nodes at 100%, 99%, 98%, 95%, and 92% detection rates. The L_8 normed NL-QGDs yielded higher SSEs than the QGD and the L_2 normed NL-QGDs after training. However, false alarms less occurred for all NL-QGDs with L_8 norm than for those of the QGD trained based the least squares. This implies that the large errors were weighted more than in the L_2 case and false alarms were therefore greatly improved. In addition, the SSE criterion is not the best choice from a detection stand point. A threshold for 100% detection was increased, thus avoiding an excessive amounts of false alarms due to a low threshold. The NL-QGDs with a large norm also show more robust performance than the NL-QGD with L_2 and QGD in the ROC curves in Figure 55a, Figure 56a, and Figure 57a.

During the training, the effective learning rate at the output layer and hidden layer [74] is modified to $\eta(n) = |e(n)|^{p-2} \bar{\eta}$ where $\bar{\eta}$ is a learning rate at $p = 2$ and $e(n)$ is the error at an iteration n at the output layer. When the errors becomes smaller as the iteration number increases, the convergence speed of the NL-QGDs base on a large norm is slow due to the $p-2$ powered pre-multiplication factor. Training iterations of the NL-QGDs based on a L_8 norm are about 10 times larger compared to the QGD for 3, 5, and 7 nodes (Table 7).

Table 8 shows the L_8 normed NL-QGD testing over the same testing sets as for the QGD and L_2 normed NL-QGD cases. The ROC curves and the outputs of the L_8 normed NL-QGDs are plotted in Figure 56 and Figure 57. The large errors for the two classes were reduced due to the larger norm (L_8).

The NL-QGDs with 3, 5, and 7 nodes yielded about 200 more false alarms than the QGD at 100% detection but less false alarms below the 100% detection than the QGD.

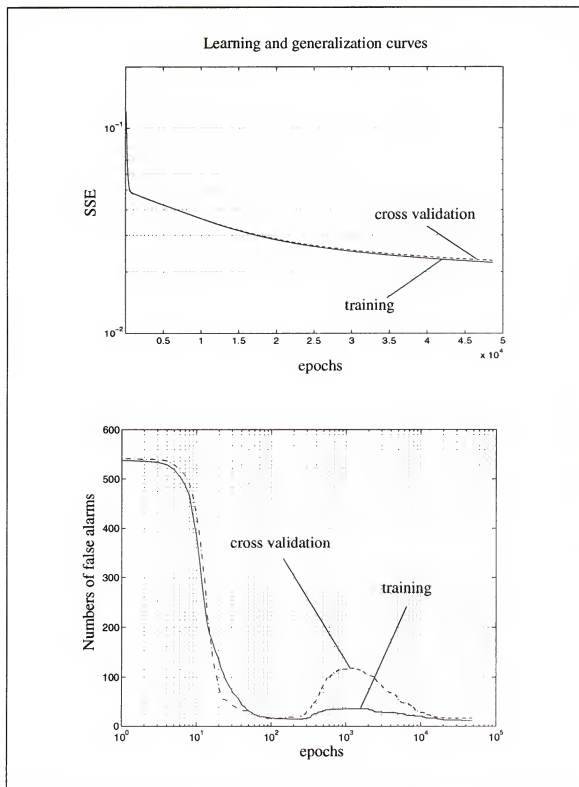


Figure 54 Learning curve and False alarm curve for 100% target detection of the L_8 normed NL-QGD751.

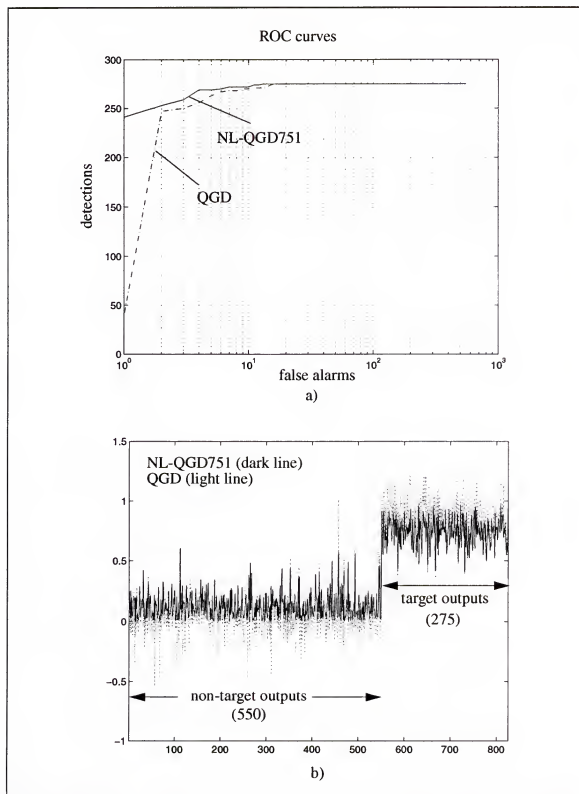


Figure 55 Training results of the L_8 normed NL-QGD751. a) ROC, b) outputs of the L_8 NL-QGD751 for the training set.

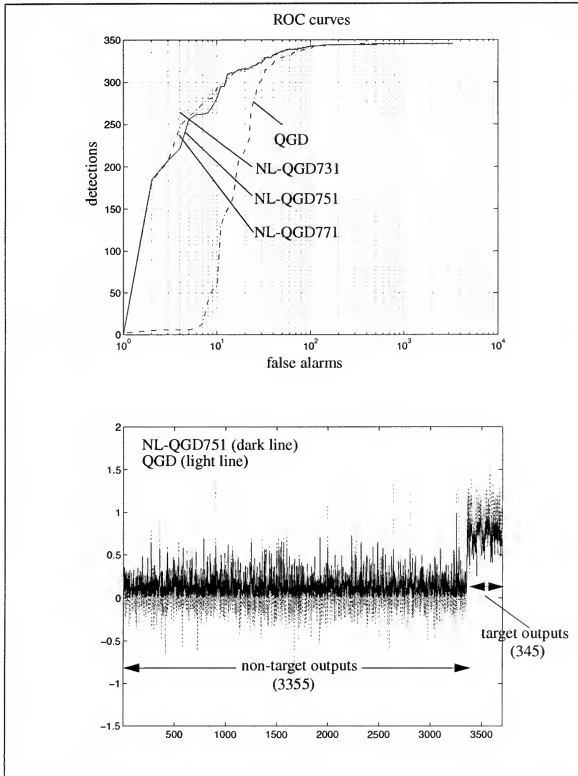


Figure 56 Testing results of the L_8 normed NL-QGD751 with 3, 5, 7 nodes for the testing set #1.2. a) ROC curves, b) outputs of the L_8 normed NL-QGD751.

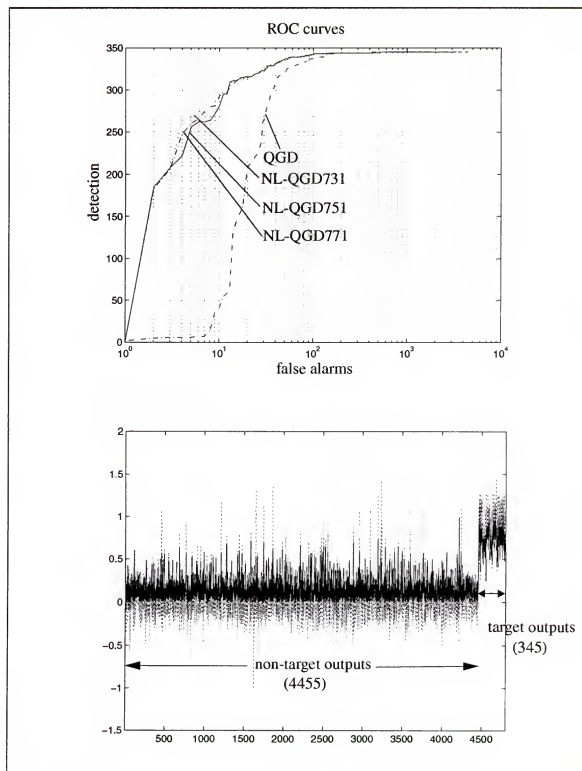


Figure 57 Testing results of the NL-QGD751 with 3, 5, 7 nodes for the testing set #2.
 a) ROC curves, b) Outputs of the NL-QGD751 with 3, 5, 7 nodes.

Table 9 Detection performance of L_8 normed NL-QGD in training set.

Network Topologies	Detection Rates					SSE	Costs **	Stop iterations
	100 %	99%	98%	95%	92%			
NL-QGD731	10	10	3	2	2	0.02168	2.550e-5	43144
NL-QGD751	12	10	5	3	1	0.02171	2.559e-5	48634
NL-QGD771	11	10	3	3	2	0.02221	2.590e-5	41819

* For NL-QGD training, the learning rate (η) and momentum rate (α) were 0.9 and 0.7 respectively for all the networks.

Table 10 Detection performance of L_8 normed NL-QGD in testing.

Network Topologies	Detection Rates (No. of false alarms in testing set #1.2/No. of false alarms in testing set #2)					SSE	Costs
	100%	99%	98%	95%	92%		
NL-QGD731	522/ 652	95/ 119	57/ 73	31/ 42	25/ 31	0.019355/ 0.018434	6.877e-5/ 6.144e-5
NL-QGD751	506/ 632	91/ 115	54/ 70	32/ 43	22/ 29	0.018682/ 0.017697	7.041e-5/ 6.114e-5
NL-QGD771	512/ 640	91/ 114	57/ 73	31/ 42	24/ 30	0.019235/ 0.018255	6.720e-5/ 5.862e-5

The L_8 normed NL-QGDs outperformed the L_2 normed NL-QGDs for all 3, 5, 7 nodes in ROC. The training performances of the L_8 normed NL-QGDs are shown to be more robust than the QGD and the L_2 normed NL-QGD cases. The NL-QGD with 3 nodes shows better detection performance than the cases with 5, 7 nodes between about 200 and 300 target detections in Figure 56a and Figure 57a but in the other operating range, the detection performances are comparable for the 3, 5, 7 node cases.

6.3.2.4 Training and testing the NL-QGD with mixed L_p norms

In order to reduce the effect of non-target outliers on the input space decision boundary formation, a small norm ($p = 1.1$) was imposed on the non-target class and a large norm ($p = 8$) on the target class. Table 11 shows the detection performance of the $L_{1.1}/L_8$ normed NL-QGDs with 3, 5, 7 nodes. A BP algorithm with momentum was used to train the $L_{1.1}/L_8$ normed NL-QGDs. The algorithm was only modified at the output layer for its local gradient according to (142). The weight adaptation was performed after presenting all training exemplars and the training was stopped when the cost in the cross validation set started increasing.

In Figure 58b, the false alarm curves for both training and cross validation sets decrease as the training iterations increase. The smaller norm ($L_{1.1}$) imposed on the non-target class emphasizes small errors and deemphasizes large errors. In Table 11, the detection performance of the $L_{1.1}/L_8$ normed NL-QGD outperformed the QGD at high probabilities of detection rates (100%, 99%, 98%, 95%, 92%) after the training.

Table 11 Detection performance of mixed norm NL-QGD in training set.

Network Topologies	Detection Rates					SSE	Costs **	Stop iterations
	100 %	99%	98%	95%	92%			
NL-QGD731	12	12	10	5	4	0.00682	0.4393	76775
NL-QGD751	12	12	10	5	4	0.00662	0.4345	82400
NL-QGD771	12	12	10	5	4	0.00664	0.4358	79747

* For NL-QGD training, the learning rate (η) and momentum rate (α) were 0.0001 and 0.00001 respectively for all the networks.

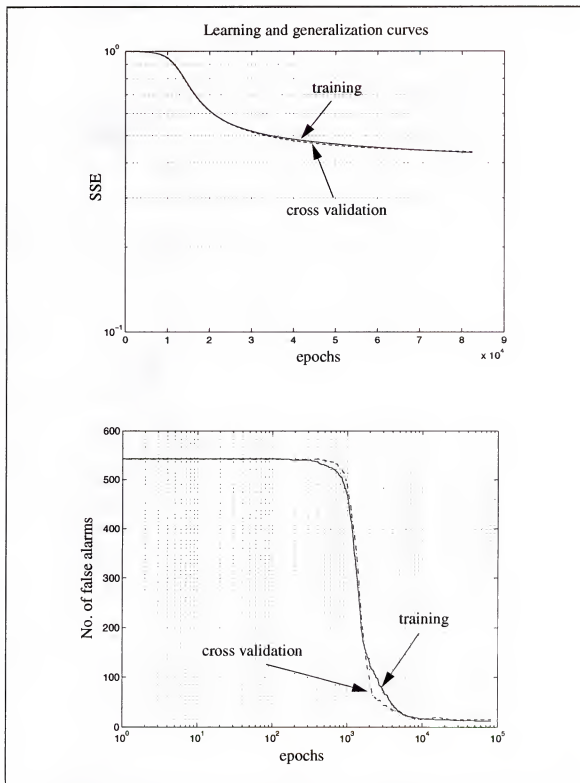


Figure 58 Learning curve and False alarm curve for 100% target detection of the $L_{1,1}/L_8$ normed NL-QGD751.

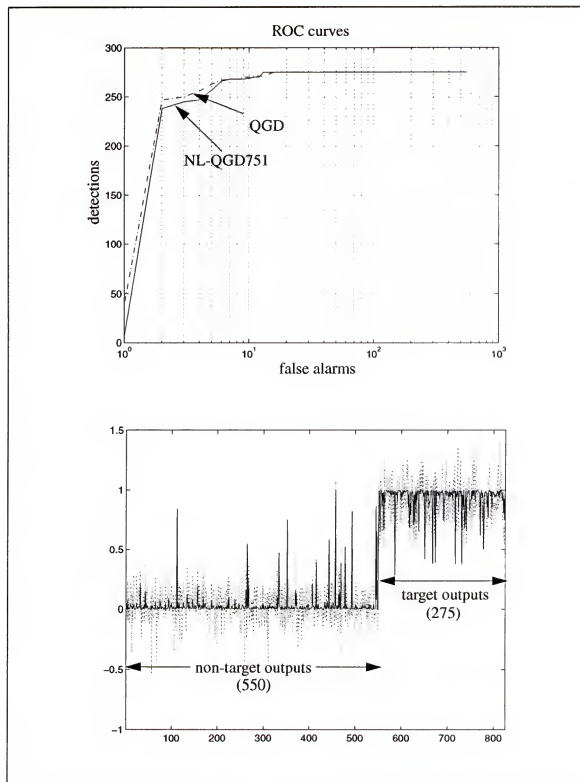


Figure 59 Training results of the $L_{1,1}/L_8$ normed NL-QGD751. a) ROC curve, b) outputs of the $L_{1,1}/L_8$ normed NL-QGD751 for the training set.

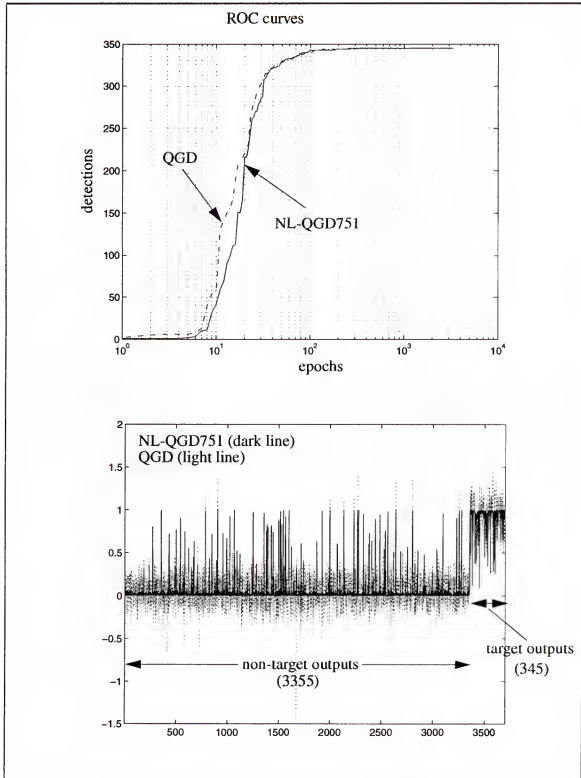


Figure 60 Testing results of the $L_{1,1}/L_8$ normed NL-QGD751 for the testing set #1.2.
 a) ROC curves, b) outputs of the $L_{1,1}/L_8$ normed NL-QGD751.

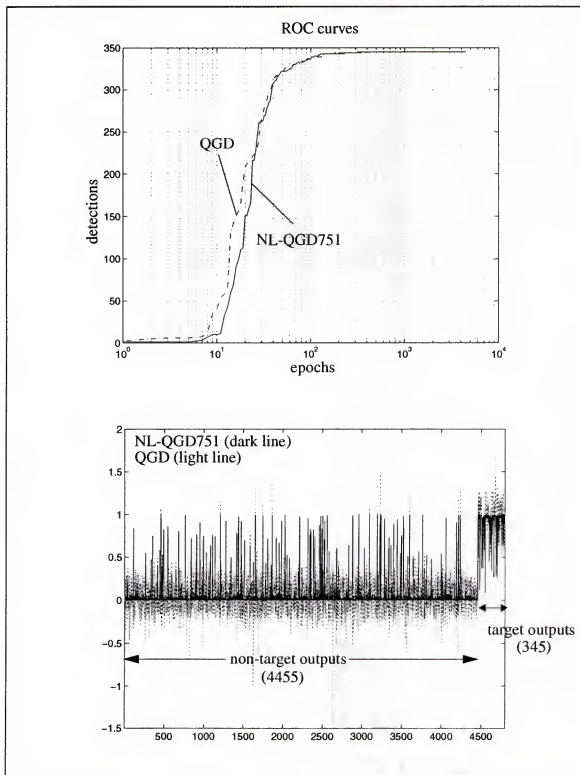


Figure 61 Testing results of the $L_{1,1}/L_8$ normed NL-QGD751 for the testing set #2. a) ROC curves, b) outputs of the $L_{1,1}/L_8$ normed NL-QGD751.

Table 12 Detection performance of mixed norm NL-QGD in testing.

Network Topologies	Detection Rates (No. of false alarms in testing set #1.2/No. of false alarms in testing set #2)					SSE
	100%	99%	98%	95%	92%	
NL-QGD731	232/ 292	97/ 124	76/ 101	40/ 63	36/ 45	0.00830/ 0.00795
NL-QGD751	252/ 316	96/ 123	76/ 101	48/ 63	36/ 45	0.00833/ 0.00800
NL-QGD771	251/ 315	96/ 123	76/ 101	48/ 63	36/ 45	0.00833 0.00799

The performance of the $L_{1,1}/L_8$ normed NL-QGDs in the testing sets is presented for different network sizes in Table 12. At the 98% ~ 100% detection range in the ROC shown in Figure 60a and Figure 61a, the $L_{1,1}/L_8$ normed NL-QGDs improved the false alarm rates compared to that of the QGD and yielded the least false alarm rate at 100% detection among the QGD and the L_2 normed and L_8 normed NL-QGDs for the three different network sizes.

Consequently, imposing a large norm on the target class moved the decision boundary towards the outliers of the target class so that the $L_{1,1}/L_8$ normed NL-QGDs did not produce large errors for the target outliers. The detection thresholds, therefore, were set to be larger at high probabilities of detection so that more non-target inputs could be discriminated. The small errors from the target inputs were more deemphasized so the detection performance became worse below a high probability (about 92%) of detection rate (Figure 60a and Figure 61a).

6.3.2.5 Training and Testing the NL-QGD with Cross Entropy

The cross entropy cost function was adapted as a network optimality index to maximize the likelihood of observing the training data set. The NL-QGDs trained based on the cross entropy put more emphasis on smaller errors and large errors than when trained by the L_p ($p > 2$) norms.

The training for the NL-QGDs was stopped at the epoch where the cross entropy function started increasing. In Figure 62b, the false alarm curves for the both training set and cross validation set decrease as the training epochs increase. As for the L_8 case, more emphasis was put on large errors than small errors. However the cross entropy function also puts more emphasis on small errors which are deemphasized in a large norm case. This leads to improving the detection performance in low probability detection ranges.

The NL-QGDs outperformed the QGD and the NL-QGDs trained L_p norms (Table 13 and Table 14). The ROCs of the NL-QGDs with cross entropy exhibited much more robust detection capability. This is compared in Figure 64a and Figure 65a.

Table 13 Detection performance of NL-QGD trained on cross entropy in training set.

Network Topologies	Detection Rates					SSE	Costs**	Stop iterations
	100%	99%	98%	95%	92%			
NL-QGD731	11	9	3	1	1	0.00409	25.12	11272
NL-QGD751	10	10	3	1	1	0.00404	24.95	11231
NL-QGD771	10	10	3	1	1	0.00405	24.88	11689

* For NL-QGD training, the learning rate (η) and momentum rate (α) were 0.9 and 0.7 respectively for all the networks.

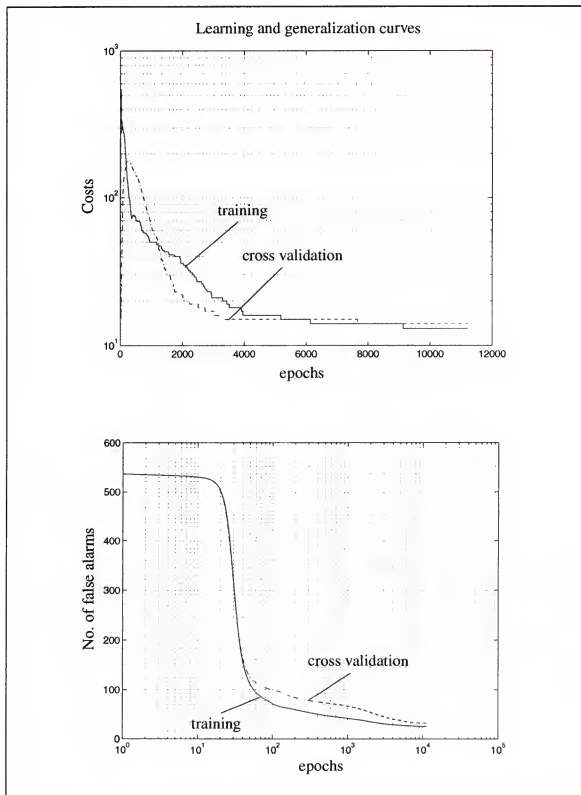


Figure 62 Learning curve and False alarm curve for 100% target detection of the NL-QGD751s trained based on the cross entropy function.

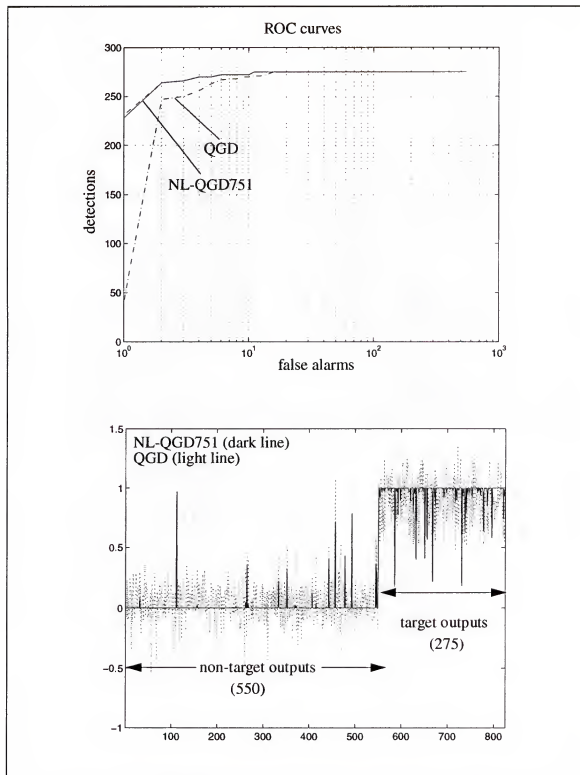


Figure 63 Training results of the NL-QGDs with a cross entropy function. a) ROC curves, b) outputs of NL-QGD751 for the training set.

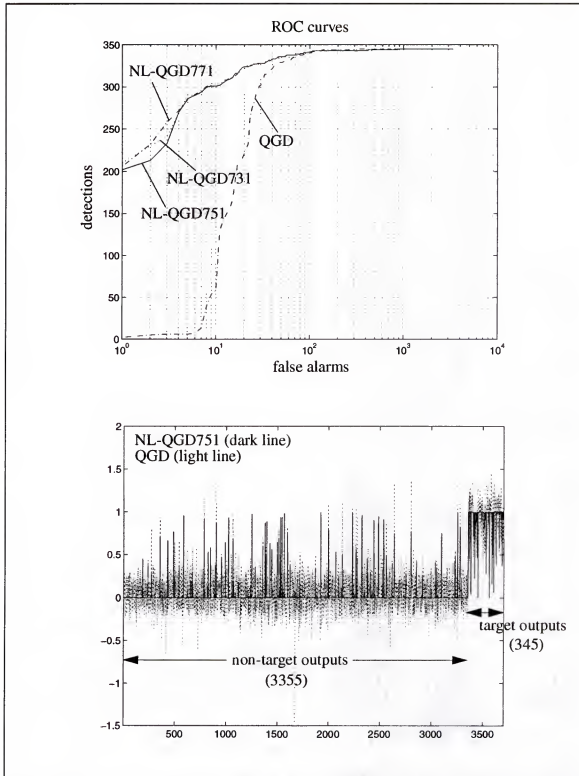


Figure 64 Testing results of the NL-QGDs with the cross entropy function for the testing set#1.2. a) ROC curves, b) outputs of the NL-QGD751 trained on the cross entropy.

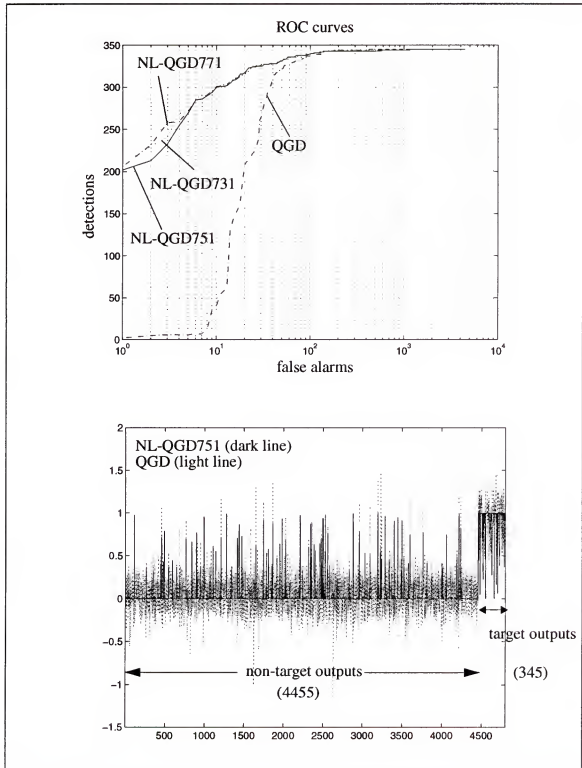


Figure 65 Testing results of the NL-QGDs with the cross entropy function for the testing set#2. a) ROC curves, b) outputs of the NL-QGD751 trained on the cross entropy.

Table 14 Detection performance of NL-QGD trained on cross entropy in testing set.

Network Topologies	Detection Rates (No. of false alarms in testing set #1./No. of false alarms in testing set #2)					SSE
	100%	99%	98%	95%	92%	
NL-QGD731	701/ 910	103/ 130	77/ 97	29/ 39	17/ 20	0.005356/ 0.004950
NL-QGD751	820/ 1064	102/ 129	70/ 90	27/ 36	17/ 20	0.005312/ 0.004959
NL-QGD771	885/ 1152	99/ 125	71/ 91	27/ 36	17/ 20	0.005309/ 0.004941

6.3.3 Summary of Detection Performance in CFAR/QGD and CFAR/NL-QGDs

The QGD reduced the number (4455) of false alarms caused by the two-parameter CFAR detector to 422 false alarms, thus achieving a discrimination power of about 1:10 (422/4455). The independent test results from Lincoln Laboratory also showed very promising discrimination using the QGD over a large data base of real-life data.

The NL-QGDs were trained and tested based on different optimality indices (L_p norm, mixed L_p norm, and cross entropy). The optimality indices influenced the detection performance of the NL-QGDs.

The L_p ($p = 2$) normed NL-QGDs produced smaller SSE than the QGD but yielded large errors for outliers. This lowered a threshold and therefore caused excessive false alarms for high detection probabilities.

The mixed norm approach was performed as an effort to reduce false alarm rates at high probabilities of detection by imposing a larger norm ($p > 2$) on the target class and a smaller norm ($p < 2$) on the non-target class. Penalizing more errors from the target class than the non-target class improved false alarms at 100% detection rates. In the training, the

non-target outliers produced large errors but these large errors were less penalized by a smaller norm. Training the NL-QGDs based on L_2 with removal of non-target outliers also effectively improved the false alarms at 100% detection. Hence these two training methods effectively handled the non-target outliers and their detection performances were superior to the QGD and NL-QGDs trained on the L_8 , L_2 , cross entropy criterions. However below 100% detection rate these two training methods (mixed norm ($L_{1,1}/L_8$) and L_2 norm with removal of non-target outliers during the training) deteriorated the detection performances of the NL-QGDs.

A large norm ($p > 2$) effectively reduced large errors and improved the false alarms compared to the L_2 normed NL-QGDs. At 100% detection, the L_8 norm still led the NL-QGDs to producing more false alarms than the QGD but improved three times the false alarms by the NL-QGDs trained on the L_2 norm. The L_8 normed NL-QGDs showed more robust detection performance below 100% detection rate.

Contrary to the L_p norms, the cross entropy function was designed to maximize the likelihood of observing the training data set regardless of the pdf assumption of the training data set. For an infinitely large data set, this indeed leads the network output to producing the *a posteriori* probability given a class input [5]. Since the outliers for each class have low probabilities (where the NL-QGDs trained on the cross entropy yielded large errors), large false alarms were produced at the range between 100% and 98% detection. However, the detection performance is most robust below 98% detection rate among the optimality indices used.

As another figure of merit for detection performance assessment, *detection robustness* is defined as the area under an ROC curve. The QGDs and the NL-QGDs trained on the L_2 norm with removal of non-target outliers and the mixed norm ($L_{1,1}/L_8$) showed that their detection performances dropped down rapidly below about 90% detection rate. Table 15 summarizes the detection performance of the QGDs and the NL-QGDs based on the number of false alarms at high probabilities of detection (100%, 99%, 98%, 95%, and 92%).

Rank	Detection rates				
	100%	99%	98%	95%	92%
11	QGD _{adapt} (695)	NL-QGD771 _{CE} *** (125)	NL-QGD731 _{CE} ** (97)	QGD _{LS} (57)	QGD _{LS} (44)
12	NL-QGD731 _{CE} (910)	NL-QGD771 _{OL} (126)	NL-QGD771 _{L1.1/L8} * (101)	NL-QGD731 _{OL} (58)	NL-QGD771 _{L1.1/L8} * (45)
13	NL-QGD751 _{CE} (1064)	NL-QGD751 _{CE} (129)	NL-QGD751 _{L1.1/L8} * (101)	NL-QGD771 _{OL} (61)	NL-QGD751 _{L1.1/L8} * (45)
14	NL-QGD771 _{CE} (1152)	NL-QGD731 _{CE} ** (130)	NL-QGD731 _{L1.1/L8} * (101)	NL-QGD751 _{OL} * (63)	NL-QGD731 _{L1.1/L8} * (45)
15	NL-QGD771 _{L2} (1880)	NL-QGD731 _{OL} ** (130)	NL-QGD751 _{OL} * (101)	NL-QGD771 _{L1.1/L8} * (63)	NL-QGD731 _{OL} * (45)
16	NL-QGD751 _{L2} (1889)	QGD _{LS} (132)	QGD _{LS} (109)	NL-QGD751 _{L1.1/L8} * (63)	NL-QGD771 _{OL} (48)
17	NL-QGD731 _{L2} (1892)	QGD _{adapt} (162)	QGD _{adapt} (122)	NL-QGD731 _{L1.1/L8} * (63)	NL-QGD751 _{OL} (49)

Table 16 compares the detection robustness among the detectors. Note that the detectors marked by the same number of asterisk (***) in each column have the same detection performances (the same numbers of false alarms).

Table 16 Robustness Rank of detection performances of the detectors (QGD and NL-QGDs).

Robustness Rank	Detectors
1	NL-QGDCE
2	NL-QGD _{L8}
3	NL-QGD _{L2}
4	QGD _{adapt}
5	QGD _{LS}
6	NL-QGD _{OL}
7	NL-QGD _{L1.1/L8}

6.3.4 γ CFAR/QGD and γ CFAR/NL-QGDs

In Section 6.2, the two prescreeners, the two-parameter CFAR and γ CFAR detectors, were run over the Mission 90 Pass 5 SAR data set and their detection performances were compared. In Section 6.3.1, the QGD and the NL-QGDs were trained and tested as false alarm reducers in the false alarm reduction stage after the two-parameter CFAR detector. In this section, the QGD and NL-QGDs trained in Section 6.3.1 and Section 6.3.2 are applied to the ROIs selected by the γ CFAR detector. Retraining the QGD and the NL-QGD is not essential in conjunction with the γ CFAR detector because many detection points (after clustering) caused by the γ CFAR detector overlap with those by the two-parameter CFAR detector (Figure 38).

The detection performances of the QGD and NL-QGDs at high probabilities of detection (100% ~ 92%) are tabulated in Table 17. In general, the NL-QGDs trained on the L₈

and $L_{1,1}/L_8$ norms showed the best detection performances for different sizes of the network (3, 5, 7 nodes) at 100% detection rate. This implies that imposing a large norm on the target class increased a 100% detection threshold so that an excessive amount of false alarms was prevented from being produced due to a low threshold. The NL-QGDs trained with removal of non-target outliers showed the second best detection performance and were followed by the QGD trained with the LS and a gradient descent method in which 334 and 399 false alarms were produced respectively at 100% detection rate. The cross entropy criterion and the L_2 norm led the NL-QGDs to yielding relatively large false alarms at 100% detection rate.

The NL-QGDs trained with removal of non-target outliers still showed a good discrimination ability at 99% detection rate while their detection performances were degraded in conjunction with the two-parameter CFAR detector. The QGD_{LS} showed the second best detection performance at 99% detection rate. The $L_{1,1}/L_8$ norm, L_8 norm and cross entropy function led the NL-QGDs to producing comparable detection performance with the QGD_{adapt} at 99% detection rate. The NL-QGDs trained on the L_2 norm produced the most false alarms at 99% detection rate. At 98% detection rate, the NL-QGDs $_{L_8}$ norm produced the best detection performance followed by the NL-QGDs $_{CE}$, NL-QGDs $_{L_2}$, NL-QGDs $_{OL}$, $QGDs_{LS}$, NL-QGDs $_{L_{1,1}/L_8}$, and $QGDs_{adapt}$. Below 95% detection rate, the NL-QGDs $_{CE}$ and NL-QGDs $_{L_8}$ produced low false alarms compared to the other norms. This is in agree with the case of the QGD and NL-QGDs in conjunction with the two-parameter CFAR detector.

6.3.5 Fast Implementation of γ CFAR/QGD and γ CFAR/NL-QGDs

Currently, the implementation of the γ CFAR and QGD requires two different optimal sets of μ_m and μ_n . The γ CFAR and QGD find their optimal μ_m and μ_n values at different locations in the parameter space for their best target discriminations against clutter. This is because the γ CFAR finds its optimal μ_m and μ_n with a restricted discriminant function

while the QGD seeks its optimal μ_m and μ_n with 8 degrees of freedom in the weights for the discriminant function. Another reason for the dissimilarity in the two optimal sets of μ_m and μ_n is from the fact that for the γ CFAR detector the highest pixel intensities of the training image chips were placed at the centers of the chips, thus resulting in a peaky g_m

Table 17 Detection performance of the QGD and NL-QGDs in conjunction with the γ CFAR detector (γ CFAR/QGD and γ CFAR/NL-QGDs).

Network Topologies	Detection Rates					SSE
	100%	99%	98%	95%	92%	
QGD _{LS}	334	100	94	74	52	0.06070
QGDadapt	399	154	108	87	73	0.04580
NL-QGD731 _{L2}	677	239	90	47	38	0.02912
NL-QGD751 _{L2}	678	239	90	47	38	0.02911
NL-QGD771 _{L2}	677	239	90	47	38	0.02910
NL-QGD731 _{OL}	237	98	93	76	57	0.05171
NL-QGD751 _{OL}	207	102	92	79	68	0.07480
NL-QGD771 _{OL}	204	98	93	78	63	0.07212
NL-QGD731 _{L8}	177	147	72	45	24	0.04623
NL-QGD751 _{L8}	160	148	71	41	23	0.04586
NL-QGD771 _{L8}	167	148	73	43	23	0.04639
NL-QGD731 _{L1.1/L8}	176	146	96	75	63	0.03937
NL-QGD751 _{L1.1/L8}	176	146	95	75	62	0.03987
NL-QGD771 _{L1.1/L8}	173	145	95	75	62	0.03977
NL-QGD731 _{CE}	451	145	88	28	23	0.02384
NL-QGD751 _{CE}	431	152	86	27	22	0.02401
NL-QGD771 _{CE}	434	182	86	27	22	0.02404

kernel ($m = 1$) while for the QGD the center points of the training image chips are centroids of detection points in the image so that the highest pixel intensities are no longer

placed at the chip centers. This led g_m kernel ($m = 1$) of the QGD to stretching out to capture high pixel values under the kernel, thus resulting in g_1 to be broader compared to that of the γ CFAR detector.

Focus of attention for ATD/R requires fast implementation of processing the input data. If the QGD uses the same μ_m and μ_n values with the γ CFAR detector the computational bandwidth of the QGD can be reduced because 5 features of $(g_m \bullet X)$, $(g_n \bullet X)$, $(g_n \bullet X)^2$, $(g_n \bullet X^2)$, and $(g_m \bullet X)(g_n \bullet X)$ are not required to be computed. This expedites the operation speed of the focus of attention stage in the multi-stage ATD/R system. However, the usage of the γ CFAR's μ_m and μ_n will degrade the detection performance of the QGD. This is shown in Figure 39a. With the γ CFAR's optimal μ_m and μ_n (μ_1 index = 22 and μ_{15} index = 15), the QGD produced 280 false alarms (5 times the number of false alarms with the QGD's optimal μ_1 and μ_{15} values). A way of obtaining a reasonable detection performance of the QGD at the γ CFAR's optimal μ_1 and μ_{15} is to train the QGD with training image chips in which the highest intensities are placed at the chip centers. This is because the QGD may capture its discrimination ability with a peakier g_1 kernel in the parameter space. Figure 66 shows the QGD false alarm surface with the same training image chips used in Figure 39a but the highest pixel intensities at the chip centers.

The minimum number of false alarms was 38 and occurred at μ_1 index 17 and μ_{15} index 14 in the parameter space. In this false alarm surface, the number of false alarms was 76 at the γ CFAR's optimal μ_1 and μ_{15} . From this training set, the QGD produced 3.5 times less false alarms than the QGD with the training set used in Figure 39a.

In conclusion, the same usage of μ_1 and μ_{15} for both the γ CFAR detector and the QGD expedites the computation speed in the focus of attention stage and does not require an exhaustive search of the parameter space so that an optimal weight set of the QGD can be obtained by the LS solution given a predetermined set of μ_1 and μ_{15} values. This is indeed a big advantage in the QGD training time and in a real-time operation of the multi-stage ATD/R system.

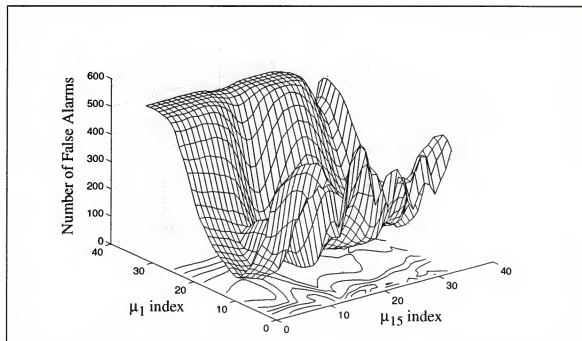


Figure 66 False alarm surface versus μ_1 and μ_{15} . Note that in this training set the image chips have the highest pixel intensities at the chip centers.

CHAPTER 7

CONCLUSIONS AND FUTURE RESEARCH

7.1 Summary

In this study, a novel approach to the design of a focus of attention stage was proposed based on gamma kernels for a multi-stage ATD/R system. The focus of attention stage is composed of two substages: a front-end detection stage and a false alarm reduction stage.

The front-end detection stage employs a prescreener to nominate potential target locations in the image. Here we discussed a common prescreener, the two-parameter CFAR detector and extended it to the γ CFAR detector. The two-parameter CFAR detector measures a target mean by a target masking kernel and clutter statistics of mean and variance by clutter masking kernel in the CFAR stencil. The γ CFAR detector incorporates a set of 2-D gamma kernels into the γ CFAR stencil which relaxes a constraint of the fixed size of the CFAR stencil. A first order gamma kernel and a higher order gamma kernel ($n > 1$) constitutes the γ CFAR stencil in which a target mean is computed by the first order gamma kernel and clutter statistics of mean and variance are estimated by the higher order gamma kernel.

The 2-D gamma kernels were extended by a circularly symmetric rotation from their 1-D counterparts. The 2-D gamma kernels preserve the 1-D characteristic of memory depth in the 2-D domain. The memory depth is controlled by the kernel order n and the parameter μ which determine the shape and the scale of the γ CFAR stencil. Thus the γ CFAR stencil can be adaptively set to better optimize a figure of merit (false alarm rates, detector output errors, etc.).

The two-parameter CFAR detector and the γ CFAR detector were interpreted in a sig-

nal processing perspective. In the CFAR stencil, a local region of image is decomposed by two local projection operators (a target masking kernel and clutter masking kernel), which may be suboptimal due to a fixed size of window. However, the γ CFAR stencil locally projects a region of image under analysis onto a gamma basis by two gamma kernels (one having $n = 1$ and the other having an order ($n > 1$)). The parameter μ provides a degree of freedom to rotate the gamma basis so that a figure of merit could be better optimized.

From the detection performance comparison in the mission 90 pass 5 SAR data set, the γ CFAR detector outperformed the two-parameter CFAR detector in each frame and yielded 760 false alarms while the two-parameter CFAR detector produced 4455 false alarms (1:6 ratio). Overall, the ROC of the γ CFAR detector exhibited more robust detection performance than the two-parameter CFAR detector.

After prescreening the entire image in the front-end detection stage, a more sophisticated algorithm is applied to the locations nominated by the prescreener to further discriminate objects of interest against clutter (false positive), in the false alarm reduction.

The two-parameter CFAR detector was interpreted in terms of pattern recognition. The two-parameter CFAR detector implements a restricted linear discriminant function of quadratic terms of image intensities. From this perspective, the two-parameter CFAR detector could be improved: (1) it uses only some of quadratic terms of image intensity on a pixel and its surrounds; (2) it implements a fixed parametric combination of these features; (3) there is little flexibility in the feature extraction because the target masking and the clutter masking in the stencil are ad-hoc.

The QGD, an extended form of the two-parameter CFAR detector, improved these three aspects by exploiting all quadratic and linear terms of two image intensities (target mean intensity and clutter mean intensity) extracted in the γ CFAR stencil and then by constructing a linear discriminant function based on the features. The optimal weights are computed in a closed form (LS solution) through an exhaustive search of the parameter space but can be adaptively found in the parameter and weight space in an iterative manner

by a gradient descent method.

Being compared to the two-parameter CFAR detector, the QGD generalizes it with respect to: (1) the shape of the kernels used for estimating the mean and variance; (2) the selection of the weights of the decision function which is not a priori but are found through optimization; (3) utilizing all quadratic and linear terms of the two intensity features (target mean and clutter mean).

So our conclusion is that not only the γ CFAR detector but also the QGD can be viable alternatives to the two-parameter CFAR detector: The γ CFAR detector can replace the two-parameter CFAR detector as a prescreener. Notice that the raw features that the QGD requires are a superset of the ones used for the γ CFAR detector. So with one preprocessor (QGD), we can implement both a detector and a false alarm reducer (or a discriminator).

The NL-QGD extended the linear structure of the QGD into a nonlinear structure as a MLP. The NL-QGD has a potential to improve the detection performance since the MLP has capable of creating arbitrary discriminant functions. Both the QGD and the NL-QGD were tested and compared based on detection performance in ROC. The QGD reduced 4455 false alarms triggered by the two-parameter CFAR detector to 422 false alarms at 100% detection rate, achieving a discrimination power (1:10 ratio). The NL-QGDs trained on L_2 norm outperformed the QGD in most range of ROC for different network sizes (3, 5, 7 nodes) but produced excessive false alarms at 100% detection rate due to the nature of a nonlinear system.

Since the NL-QGD is designated to operate on high probabilities of detection, the effort has been made on improving the false alarms at high probability rates. By incorporating a higher norm ($p > 2$) into the NL-QGD cost function, the NL-QGD trained on L_8 produced 665, 632, and 640 false alarms at 100% detection rate. These false alarms still outnumber the false alarms caused by the QGD but the L_8 normed NL-QGDs greatly improved the performance below 100% detection rate. The detection performance of the NL-QGD with mixed norms was improved at 100% detection rate (292, 316, 315 false

alarms for 3, 5, 7 nodes) and became inferior but comparable to the QGD below 98% detection rate. The cross entropy function was finally used as an optimal index for the NL-QGD. The NL-QGDs trained on the cross entropy function showed more robust performance compared to the QGD, the L_2 normed NL-QGDs, the L_8 normed NL-QGDs, and the NL-QGD with the mixed norms ($L_{1,1}$ for non-target class and L_8 for target class). With about 30% loss of targets, they were able to completely reject false alarms.

7.2 Future works

The γ CFAR stencil relaxed the constraint of a fixed size of the CFAR stencil. Even the γ CFAR stencil can be adaptively set to optimize a figure of merit the kernel orders were a priori determined to cover a full scattering range of known targets.

However, multi-scale objection detection may face the range uncertainties of different targets and the differences between the size shapes at target types projected over all aspects and elevations. In order to solve the expected problems, it may be necessary to utilize a full set of gamma kernels to cover the ranges at possible situations. Potential targets may be declared by the highest scores measured over the entire set of gamma kernels.

The use of the full gamma kernel size for image analysis is left for further research. Also, the determination of the optimal order to estimate the background mean and variance has not been addressed yet, since there is a quasi-symmetry between a change in order and a change in μ .

The projection on the circularly symmetric 2-D gamma kernels are complete so that a input image can be recovered from the projection. Extending 1-D gamma kernels to 2-D gamma kernels on which the projection could be complete will allow for image reconstruction from a projection. The image analysis can be accomplished by the 2-D gamma kernels which can be pdfs.

APPENDIX

POLARIZATION BASIS TRANSFORMATION

A.1 Representation of a Plane Wave Polarization

A plane wave E , propagating in the $+z$ -direction can be express as

$$E = (E_x \hat{u}_x + E_y \hat{u}_y) e^{-kz} \quad (153)$$

In (153), both E_x and E_y are complex and can be written for a time-varying plane wave as

$$E_x = E_{ox} e^{j\omega t}, \quad E_y = E_{oy} e^{j(\omega t - \epsilon)} \quad (154)$$

where ϵ is the relative phase difference between E_x and E_y , both of which are travelling in the $+z$ -direction. The time-varying plane wave \underline{E} is expressed as

$$\begin{aligned} \underline{E} &= \text{Re} \{ E \} \\ &= E_{ox} \cos(\omega t - kz) \hat{u}_x + E_{oy} \cos(\omega t - kz - \epsilon) \hat{u}_y \\ &= \underline{E}_x \hat{u}_x + \underline{E}_y \hat{u}_y \end{aligned} \quad (155)$$

where

$$\underline{E}_x = E_{ox} \cos(\omega t - kz) \quad (156)$$

$$\underline{E}_y = E_{oy} \cos(\omega t - kz - \epsilon). \quad (157)$$

Expanding the expression for E_y into

$$\underline{E}_y / E_{oy} = \cos(\omega t - kz) \cos(\epsilon) - \sin(\omega t - kz) \sin(\epsilon)$$

and combining it with \underline{E}_x / E_{ox} yields

$$\frac{\underline{E}_y}{E_{oy}} - \frac{\underline{E}_x}{E_{ox}} \cos(\epsilon) = -\sin(\omega t - kz) \sin(\epsilon) \quad (158)$$

It follows from (156) that

$$\sin(\omega t - kz) = \left[1 - \left(\frac{\mathcal{E}_x}{E_{ox}} \right)^2 \right]^{1/2}$$

so (158) leads to

$$\left(\frac{\mathcal{E}_y}{E_{oy}} - \frac{\mathcal{E}_x}{E_{ox}} \cos(\epsilon) \right)^2 = \left[1 - \left(\frac{\mathcal{E}_x}{E_{ox}} \right)^2 \right] \sin^2(\epsilon)$$

By rearranging the above expression, we have

$$\left(\frac{\mathcal{E}_y}{E_{oy}} \right)^2 + \left(\frac{\mathcal{E}_x}{E_{ox}} \right)^2 - 2 \left(\frac{\mathcal{E}_y}{E_{oy}} \right) \left(\frac{\mathcal{E}_x}{E_{ox}} \right) \cos(\epsilon) = (\sin(\epsilon))^2 \quad (159)$$

This is the equation of an ellipse making an angle α with the (\hat{u}_x, \hat{u}_y) -coordinate system such that

$$\tan(2\alpha) = \frac{2E_{ox}E_{oy}\cos(\epsilon)}{E_{ox}^2 - E_{oy}^2} \quad (160)$$

and the wave (153) is then said to be elliptically polarized. The ellipse of (159) is graphically described in Figure 67.

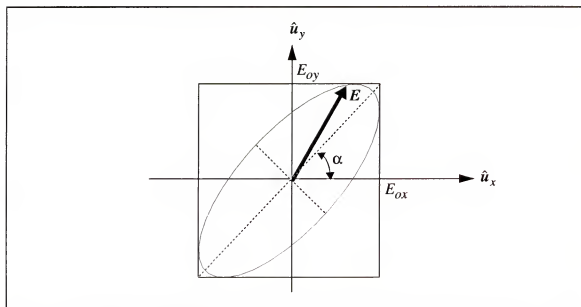


Figure 67 Graphical description of an elliptically polarized wave.

A particular wave can be stated in terms of its specific state of polarization. The polarization state includes linear polarization, and right- or left-circular polarization. So the condition of elliptic polarization corresponds to a polarization state. The polarization state is defined by the following parameters

- Linear polarization

$$\epsilon = 2\pi m \quad \text{where } m = 0, \pm 1, \pm 2, \dots$$

$$E = E_{ox} \cos(\omega t - kz) \hat{u}_x + E_{oy} \cos(\omega t - kz) \hat{u}_y \quad (161)$$

- Right-circular polarization

$$\epsilon = -\frac{\pi}{2} + 2\pi m \quad \text{where } m = 0, \pm 1, \pm 2, \dots$$

$$E_{ox} = E_{oy} = E_o$$

$$E = E_o (\cos(\omega t - kz) \hat{u}_x + \sin(\omega t - kz) \hat{u}_y) \quad (162)$$

- Left-circular polarization

$$\epsilon = \frac{\pi}{2} + 2\pi m \quad \text{where } m = 0, \pm 1, \pm 2, \dots$$

$$E_{ox} = E_{oy} = E_o$$

$$E = E_o (\cos(\omega t - kz) \hat{u}_x - \sin(\omega t - kz) \hat{u}_y) \quad (163)$$

A linearly polarized wave can be synthesized from two oppositely polarized circular waves of equal amplitude. In particular, if we add the right-circular wave of (162) to the left-circular wave of (163), we get

$$E = 2E_o \cos(\omega t - kz) \hat{u}_x \quad (164)$$

which has a constant amplitude vector of $2E_o \hat{u}_x$ and is therefore linearly polarized.

A.2 Circular-to-Linear and Linear-to-Circular Polarization Basis Transformations

If we use the x- and y-axis as a linear polarization basis for the horizontally (H) and vertically (V) time-varying components of a plane wave respectively and write the equation of a electric field (E) propagating in the +z-direction at the transmit antenna, we get

the following expression by [51]

$$\begin{bmatrix} E_H^t \\ E_V^t \end{bmatrix} = j \frac{Z_o I}{2\lambda r} \begin{bmatrix} \hat{u}_H \\ \hat{u}_V \end{bmatrix} \quad (165)$$

where Z_o is the intrinsic impedance of free space, I the current into the transmit antenna terminals, λ the wave length of the transmitted electric field, and r is a distance from the transmit antenna. If it is assumed that the fully polarimetric scattering matrix ($S^{(L)}$) of a target for a linear polarized transmit/receive antenna configuration is known, then we can write the reflected wave at the receive antenna as

$$\begin{bmatrix} E_H^r \\ E_V^r \end{bmatrix} = \frac{1}{\sqrt{4\pi r}} \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \begin{bmatrix} E_H^t \\ E_V^t \end{bmatrix} \quad (166)$$

where

$$S^{(L)} = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \quad (167)$$

The reflective wave at the receive antenna can also be written with the fully polarimetric circular scattering matrix ($S^{(C)}$) of the target in left and right circular component form which leads to the following expression

$$\begin{bmatrix} E_R^r \\ E_L^r \end{bmatrix} = \frac{1}{\sqrt{4\pi r}} \begin{bmatrix} S_{RR} & S_{RL} \\ S_{LR} & S_{LL} \end{bmatrix} \begin{bmatrix} E_R^t \\ E_L^t \end{bmatrix} \quad (168)$$

where

$$S^{(C)} = \begin{bmatrix} S_{RR} & S_{RL} \\ S_{LR} & S_{LL} \end{bmatrix} \quad (169)$$

The electric field (E) can be expressed in the circular polarization basis in right and left circular component form as

$$\begin{aligned}
E &= E_H \hat{u}_H + E_V \hat{u}_V \\
&= E_R \hat{\omega}_R + E_L \hat{\omega}_L
\end{aligned} \tag{170}$$

where E_R and E_L indicate the complex right and left circular field components respectively. According to (161), (162) and (162), the circular basis ($\hat{\omega}_R$ and $\hat{\omega}_L$) are related to the linear basis by

$$\begin{aligned}
\hat{\omega}_R &= \hat{u}_H - j\hat{u}_V \\
\hat{\omega}_L &= \hat{u}_H + j\hat{u}_V
\end{aligned} \tag{171}$$

or

$$\begin{bmatrix} \hat{\omega}_R \\ \hat{\omega}_L \end{bmatrix} = \begin{bmatrix} 1 & -j \\ 1 & j \end{bmatrix} \begin{bmatrix} \hat{u}_H \\ \hat{u}_V \end{bmatrix} \tag{172}$$

By substituting (171) into (170), The right- and left-circular components of the wave are expressed in terms of the horizontal and vertical components

$$\begin{aligned}
E_R &= \frac{1}{2} (E_H + jE_V) \\
E_L &= \frac{1}{2} (E_H - jE_V)
\end{aligned} \tag{173}$$

We substitute (173) into (168) for both the transmitted and received wave but due to the direction change in the propagation and the right-handed coordinate system after reflection against the target, we have the coordinate system ($-\hat{u}_Z, \hat{u}_H, -\hat{u}_V$) with signs changed accordingly. So we replace E_V by $-E_V$ for the reflective wave in (168).

$$\begin{bmatrix} E_H^r - jE_V^r \\ E_H^r + jE_V^r \end{bmatrix} = \frac{1}{\sqrt{4\pi r}} \begin{bmatrix} S_{RR} & S_{RL} \\ S_{LR} & S_{LL} \end{bmatrix} \begin{bmatrix} E_H^t + jE_V^t \\ E_H^t - jE_V^t \end{bmatrix} \tag{174}$$

Rearranging (174) yields the following equation

$$\begin{bmatrix} E_H^r \\ E_V^r \end{bmatrix} = \frac{1}{\sqrt{4\pi r}} \begin{bmatrix} \frac{1}{2} (S_{RR} + S_{RL} + S_{RL} + S_{RL}) & \frac{1}{2} j (S_{RR} - S_{RL} + S_{LR} - S_{LL}) \\ \frac{1}{2} j (S_{RR} + S_{RL} - S_{LR} - S_{LL}) & \frac{1}{2} (-S_{RR} + S_{RL} + S_{LR} - S_{LL}) \end{bmatrix} \begin{bmatrix} E_H^t \\ E_V^t \end{bmatrix} \tag{175}$$

By equating (175) and (166), we obtain the following relationship between the linear and circular scattering matrix elements of the target.

$$\begin{aligned}
 S_{HH} &= \frac{1}{2} (S_{RR} + S_{RL} + S_{RL} + S_{RL}) \\
 S_{HV} &= \frac{1}{2}j (S_{RR} - S_{RL} + S_{LR} - S_{LL}) \\
 S_{VH} &= \frac{1}{2}j (S_{RR} + S_{RL} - S_{LR} - S_{LL}) \\
 S_{VV} &= \frac{1}{2} (-S_{RR} + S_{RL} + S_{LR} - S_{LL})
 \end{aligned} \tag{176}$$

$$\begin{aligned}
 S_{RR} &= \frac{1}{2} (S_{HH} - jS_{HV} - jS_{VH} - S_{VV}) \\
 S_{RL} &= \frac{1}{2} (S_{HH} + jS_{HV} - jS_{VH} + S_{VV}) \\
 S_{LR} &= \frac{1}{2} (S_{HH} - jS_{HV} + jS_{VH} + S_{VV}) \\
 S_{LL} &= \frac{1}{2} (S_{HH} + jS_{HV} + jS_{VH} - S_{VV})
 \end{aligned} \tag{177}$$

REFERENCES

- [1] Baum, E. B., and Haussler, D., "What size net gives valid generalization?," *Neural Computation* **1** (1), pp. 151-160, 1989.
- [2] Baum, E. B., and Wilczek, F., "Supervised learning of probability distributions by neural networks, In Anderson, D. Z. (Ed.), *Neural Information Processing Systems*, pp. 52-61, 1988.
- [3] Bernardon, A. M., and Carrick, J. E., "A neural system for automatic target learning and recognition applied to bare and camouflaged SAR targets," *Neural Networks*, vol. 8, no. 7/8, pp. 1103-1108, 1995.
- [4] Bhanu, B., "Automatic target recognition: state of the art survey," *IEEE Transactions on Aerospace and Electronic Systems*, AES-22, no. 4, pp. 364-379, 1986.
- [5] Bishop, C. M., *Neural Networks for Pattern Recognition*. Oxford University Press Inc., New York, 1995.
- [6] Born, M., and Wolf, E., *Principles of Optics*, Pergamon Press, New York, 1959.
- [7] Burrascano, P., "A norm selection criterion for the generalized delta rule," *IEEE Transactions on Neural Networks*, vol. 2, no. 1, 1991.
- [8] Duda, R. O., and Hart, P. E., *Pattern Classification and Scene Analysis*, John Wiley & Sons, 1973.
- [9] Burl, M. C., Owirka, G. J., and Novak, L. M., "Texture discrimination in synthetic aperture radar imagery," *23rd Asilomar Conf. on Signals, Systems, and Computers*, Pacific Cove, CA, 30 Oct.-1 Nov., pp. 399-404, 1989.
- [10] Burl, M. C., and Novak, L. M., "Polarimetric segmentation of SAR imagery," *Proceedings of the SPIE, Automatic Object Recognition*, vol. 1417, pp. 92-115, 1991.
- [11] Casasent, D. P., and Neiberg, L. M., "Classifier and shift-invariant automatic target recognition neural networks," *Neural Networks*, vol. 8, no. 7/8, pp. 1117-1129, 1995.
- [12] Cathcart, J. M., Doll, T. J., and Schmieder, D. E., "Target detection in urban clutter," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1242-1250, 1989.

- [13] Celebi, S. and Principe, J., "Parametric signal representation using gamma bases," *IEEE Trans. on Signal Processing*, vol. 43, no. 3, pp. 781-784, 1995.
- [14] Chaney, R. D., Burl, M. C., and Novak, L. M., "On the performance of polarimetric target detection algorithms," *Record of the 1990 IEEE International Radar Conference*, pp. 520-525, Arlington, VA, 1990.
- [15] Chang, C., Siu, S., and Wei, C., "On the convergence of the L_p norm algorithm for polynomial perceptron having different error signal distributions," *Journal of the Chinese Institute of Engineers*, vol. 18, no. 2, pp. 293-302, 1995.
- [16] Curlander, J. C., and McDonough R. N., *Synthetic Aperture Radar Systems and Signal Processing*. John Wiley & Sons Inc., New York, 1991.
- [17] Currie, A., "Synthetic aperture radar," *Electronics & Communication Engineering Journal*, Aug., pp. 159-170, 1991.
- [18] Daniell, C. D., Kemsley, D. H., Lincoln, W. P., Tackett, W. A., and Baraghimian, G. A., "Artificial neural networks for automatic target recognition," *Optimal Engineering*, vol. 31, no. 12, pp. 2521-2531, 1992.
- [19] de Vries B. and Principe, J., "The gamma model-A new neural model for temporal processing," *Neural Networks*, vol. 5, pp. 565-576, 1992.
- [20] de Vries B., "Temporal processing with neural networks: The development of the gamma model," Ph.D., dissertation, University of Florida, 1991.
- [21] Delaney, R. L., Verly J. G., and Dudgeon D. E., "Automatic building and supervised discrimination learning of appearance models of 3-D objects," *Proceedings of the SPIE*, vol. 1708, pp. 549-560, 1992.
- [22] Duda, Richard O., *Pattern Classification and Scene Analysis*, John Wiley & sons, New York, 1973.
- [23] Dudgeon, D. E., and Lacoss, R. T., "An overview of automatic target recognition," *Linc. Lab. J.*, vol. 6, no. 1, pp. 3-10, 1993.
- [24] Fitch, J. P., *Synthetic Aperture Radar*. Springer-Verlag, New York, 1988.
- [25] Fogler, R. J., Hostetler, L. D., and Hush, D. R., "SAR clutter suppression using probability density skewness," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 30, no. 2, pp. 622-626, 1994.
- [26] Forman, A. V., Jr., "Systolic Implementation of Viewpoint Invariant Automatic Target Recognition," Ph.D. dissertation, University of Florida, 1992.
- [27] Fukunaga, K. *Introduction to Statistical Pattern Recognition* Boston. Academic Press, New York, 1990.

- [28] Geman, S. Bienenstock, E., and Doursat, R., "Neural Networks and the bias/variance dilemma," *Neural Computation*, 4 (1), pp. 1-58, 1992.
- [29] Goldstein, G. B., "False-alarm regulation in log-normal and Weibull clutter," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-9, no. 1, pp. 84-92, Jan. 1973.
- [30] Golub, G. H. and Van Loan, C. F., *Matrix Computations*. Johns Hopkins, Baltimore, 1983.
- [31] Hanson, S. J., and Burr, D. J., "Minkowski-r back-propagation: learning in connectionist models with non-Euclidean error signals," In D. Anderson (Ed.) *Neural Information Processing Systems*, pp. 348-357, 1988.
- [32] Haykin, S., *Adaptive Filter Theory*, 2nd ed. Prentice-Hall, Englewood Cliffs, NJ, 1991.
- [33] Haykin, S., *Neural Networks*, Macmillan College Publishing Co., New York, 1994.
- [34] Hecht, E., *Optics*, Addison-Wesley Publishing Co., Reading, MA, 1987.
- [35] Hecht-Nielsen, R., *Neurocomputing*, Addison-Wesley Publishing Co., 1990.
- [36] Hinton, G. E., "Connectionist learning procedures," *Artificial Intelligence*, vol. 40, pp. 185-234, 1989.
- [37] Honik, K., Stinchcombe, M., and White, H., "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359-366, 1989.
- [38] Honik, K., "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251-257, 1991.
- [39] Hopfield, J. J., "Learning algorithms and probability distributions in feed-forward and feed-back networks," *Proceedings of the National Academy of Sciences*, vol. 84, pp. 8429-8433, 1987.
- [40] Kim, M, Fisher, J, and Principe, J. C., "A new CFAR stencil for target detections in Synthetic Aperture Radar (SAR) imagery," *Proceedings of the SPIE - The International Society for Optical Engineering, (Algorithms for Synthetic Aperture Radar Imagery III)*, vol. 2757, pp. 432-442, April, 1996.
- [41] Kim, M. and Principe, J., "Artificial neural networks with gamma kernels for automatic target detection," *Proceedings of the International Conference on Neural Networks*, Washington, DC, vol. 3, pp. 1594-1599, 1996.
- [42] Koch, M. W., Moya, M. M., Hostetler, L. D., and Fogler, R. J., "Cueing, Feature Discovery, and One-class learning for synthetic aperture radar automatic target recognition," *Neural Networks*, vol. 8, no.7/8, pp. 1081-1102, 1995.

- [43] Konstantinides, K. and Rasure, J. R., "The Khoros software development environment for image and signal processing," *IEEE Transactions on Image Processing*, vol. 3, no. 3, pp. 243-252, 1994.
- [44] Kraus, J. D., *Antennas*. McGraw-Hill Book Co., New York, 1950.
- [45] Kreithen, D. E., Halversen, S. D., and Owirka, G. J., "Discriminating targets from clutter," *Linc. Lab. J.*, vol. 6, no. 1, pp. 25-52, 1993.
- [46] Levanon, N., *Radar Principles*. John Wiley & Sons Inc., New York, 1988.
- [47] Liu, J., and Chang, K., "Feature-based target recognition with a Bayesian network," *Optical Engineering*, vol. 30, no. 3, pp. 701-707, 1996.
- [48] Lim, J. S., *Two-dimensional Signal and Image Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1990.
- [49] Miller, K. S., *Complex Stochastic Processes*, Addison-Wesley Publishing Co., Reading, MA, 1974.
- [50] Money, A. H., Affleck-Graves, J. F., Hart, M. L., and Barr, G. D. I., "The Linear Regression Model: L_p norm estimation and the choice of p ," *Communications in Statistics: Simulation and Computation*, vol. 11, no. 1, pp. 89-109, 1982.
- [51] Mott, H., *Polarization in Antennas and Radar*. John Wiley & Sons Inc., New York, 1986.
- [52] Novak, L. M., Sechtin, M. B., and Cardullo, M. J., "Studies of target detection algorithms that use polarimetric radar data," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-25, no. 2, pp. 150-165, Mar. 1989.
- [53] Novak, L. M., and Burl, M. C., "Optimal speckle reduction in polarimetric SAR imagery," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-26, no. 2, pp. 293-305, Mar. 1990.
- [54] Novak, L. M., Burl, M. C., Chaney R., and Owirka, G. J., "Optimal processing of polarimetric synthetic aperture radar imagery," *Linc. Lab. J.*, vol. 3, no. 2, pp. 273-290, 1990.
- [55] Novak, L. M., and Netishen, C. M., "Polarimetric synthetic aperture radar imaging," *International Journal of Imaging Systems and Technology*, vol. 4, pp. 306-318, 1992.
- [56] Novak, L. M., Burl, M. C. and Irving, W. W., "Optimal polarimetric processing for enhanced target detection," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-29, no. 1, pp. 234-244, Jan. 1993.
- [57] Novak, L. M., Owirka, G. J., and Netishen, C. M., "Performance of a high-resolution polarimetric SAR automatic target recognition system," *Linc. Lab. J.*, vol. 6, no. 1, pp.

11-24, 1993.

- [58] Novak, L. M., Halversen, S. D., Owirka, G. J., and Hielt, M., "Effects of polarization and resolution on the performance of a SAR automatic target recognition," *Linc. Lab. J.*, vol. 8, no. 1, pp. 49-65, 1995.
- [59] Oppenheim, A., and Schafer, R., *Digital Signal Processing*. Prentice-Hall, Englewood Cliffs, 1975.
- [60] Owirka, G. J., and Noval, L. M., "A new SAR ATR algorithm suit," *Proceedings of the SPIE - The International Society for Optical Engineering, (Algorithms for Synthetic Aperture Radar)*, vol. 2230, pp. 336-343, 1994.
- [61] Pao, Yoh-Han, *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley Publishing Co., Reading, MA, 1989.
- [62] Pei, S., and Tseng, C., "Least Mean P-Power Error Criterion for Adaptive FIR Filter," *IEEE Transactions on Selected Areas in Communications*, vol. 12, no. 9, pp. 1540-1547, 1994.
- [63] Principe, J. C., A. Radisavljevic, M. Kim, J. Fisher III, M. Hielt, and L. M. Novak, "Target Prescreening Based on 2D Gamma Kernels," *Proceedings of the SPIE - The International Society for Optical Engineering, (Algorithms for Synthetic Aperture Radar Imagery II)*, vol. 2487, pp. 251-258, FL, USA, April, 1995.
- [64] Principe, J. C., deVries, B., and de Oliveira, P. G., "The gamma filter-A new class of adaptive IIR filters with restrictive feedback," *IEEE Transactions on Signal Processing*, vol. 41, no. 2, pp. 649-656, 1993.
- [65] Principe J.C., Kuo J.M., and Celebi S., "An analysis of the gamma memory in dynamic neural networks," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 331-337, 1994.
- [66] Rosenfeld, A., and Thurston, M., "Edge and curve detection for visual scene analysis," *IEEE Transactions on Computers*, vol. C-20, pp. 562-569, 1971.
- [67] Roth, M. W., "Survey of neural network technology for automatic target recognition," *IEEE Transactions Neural Networks*, vol. 1, no. 1, pp. 28-43, Mar 1990.
- [68] Rumelhart, D. E., Hinton, G. E., and Williams, "Learning internal representations by error propagation," In *Parallel Distributed Processing*, vol. 1, Chapter8, MIT press, 1986.
- [69] Rumelhart, D. E., Durbin, R., Golden, R., and Chauvin, Y., "Backpropagation: the basic theory," In Chauvin, Y., and Rumelhart, D. E. (Eds.), *Backpropagation: Theory, Architecture, and Applications*, pp. 1-34, Lawrence Erlbaum, Hillsdale, NJ, 1995.
- [70] Ruzinsky, S. A., and Olsen, E., " L_1 and L_∞ minimization via a variant of Karmarkar's

- algorithm," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, no. 2, pp. 245-253, 1989.
- [71] Schachter, "A survey and evaluation of FLIR target detection/segmentation algorithms," *Proceedings of Image Understanding Workshop*, pp. 49-57, 1982.
- [72] Schmieder, D. E., and Weathersby, M. R., "Detection performance in clutter with variable resolution," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-19, no. 4, pp. 622-630, 1983.
- [73] Schmieder, D. E., and Weathersby, M. R., "Developing texture-based image clutter measures for object detection," *Optimal Engineering*, vol. 31, no. 12, pp. 2628-2639, 1992.
- [74] Schroeder, J., Yarlagadda, R., and Hershey, J., "Lp normed minimization with applications to linear predictive modeling for sinusoidal frequency estimation," *Signal Processing*, vol. 24, pp. 193-216, 1991.
- [75] Schroeder, J., and Endsley, J., "Lp normed spectral estimation residual analysis," *1988 22nd Asilomar Conference on Signal, Systems and Computers*, pp. 848-852, 1988.
- [76] Shirvaikar, M. V., and Trivedi, M. M., "A neural network filter to detect small targets in high clutter backgrounds," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 252-257, 1995.
- [77] Siu, S., Chang, C. H., and Wei, C. H., "Lp norm back propagation algorithm for adaptive equalization," *IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 42, no. 9, pp. 604-607, 1995.
- [78] Siu, S., and Cowan, C. F. N., "Performance analysis of the lp norm back propagation algorithm for adaptive equalisation," *IEE Processing-F*, vol. 140, no. 1, pp.43-47, 1993.
- [79] Solla, S. A., Levin, E., and Fleisher, M., "Accelerated learning in layered neural networks," *Complex Systems*, vol. 2, pp. 625-640, 1988.
- [80] Stone, M., "Cross-validatory choice and assessment of statistical predictions," *Journal of the Royal Statistical Society*, vol. B 36, no. 1, pp. 111-147, 1974.
- [81] Tou, J. T., and Gonzalez, R. C., *Pattern Recognition Principles*. Addison-Wesley Publishing Co., Reading, MA, 1974.
- [82] Ukrainec A. M., and Haykin, S., "A modular neural network for enhancement of cross-polar radar targets," *Neural Networks*, vol. 9, no. 1, pp. 143-168, 1996.
- [83] Verbout, S. M., Irving, W. W., and Hanes, A. S., "Improving a template-based classifier in a SAR automatic target recognition system by using 3-D target information,"

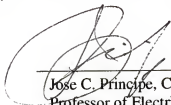
Linc. Lab. J., vol. 6, no. 1, pp. 53-76, 1993.

- [84] Verly, J. G., Delanoy, R. L., and Dudgeon, D. E., "Model-based system for automatic target recognition from forward-looking laser-radar imagery," *Optical Engineering*, vol. 31, no. 12, pp. 2540-2552, 1992.
- [85] Verly, J. G., Delanoy, R. L., Lazott, C. H., and Dudgeon, D. E., "A Model-based Automatic target recognition system for Synthetic Aperture Radar Imagery," *2nd Automatic Target Recognizer Systems and Technology Conference*, pp. 51-80, 1992.
- [86] Verly, J. G., and Dudgeon, D. E., "Model-based automatic target recognition system for the UGV/RSTA radar," *Image Understanding Workshop*, pp. 559-583, Nov., 1994.
- [87] Waxman, A. M., Seibert, M. C., Gove, A., Fay, D. A., A. M., Lazott, C., Steele, W. R., and Cunningham, R. K., "Neural Processing of Targets in Visible, Multispectral IR and SAR Imagery," *Neural Networks*, vol. 8, no. 7/8, pp. 1029-1051, 1995.
- [88] Waxman, A. M., Lazott, C., Fay, D. A., Gove, A., and Steele, W. R., "Neural Processing of SAR Imagery for enhanced target detection," *Proceedings of the SPIE, Algorithms for Synthetic Aperture Radar Imagery II*, vol. 2487, pp. 201-210, 1995.
- [89] Widrow, B., and Stearns, S. D., *Adaptive Signal Processing*. Prentice-Hall Inc., Englewood Cliffs, NJ, 1985.
- [90] Yarlagadda, R., Bednar, J. B., and Watt, T. L., "Fast algorithms for l_p deconvolution," *IEEE Transactions on Acoustic, Speech, and Signal Processing*, vol. ASSP-33, no. 1, pp. 174-182, 1985.

BIOGRAPHICAL SKETCH

Munchurl Kim was born in Korea on January 25th, 1966. He received his Bachelor of Engineering degree in electronics from the KyungPook National University, Korea, in 1989. After graduation he joined the graduate program in electrical and computer engineering at the University of Florida in which he received a Master of Engineering degree in 1992 and a Ph.D. in 1996. He plans to continue conducting research in the areas of automatic target detection/recognition, digital signal and image processing, pattern recognition, and artificial neural networks.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in score and quality, as a dissertation for the degree of Doctor of Philosophy.



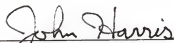
Jose C. Principe, Chairman
Professor of Electrical and Computer
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in score and quality, as a dissertation for the degree of Doctor of Philosophy.



William W. Edmonson
Assistant Professor of Electrical and
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in score and quality, as a dissertation for the degree of Doctor of Philosophy.



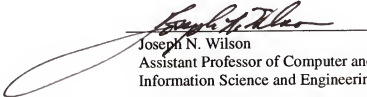
John G. Harris
Assistant Professor of Electrical and
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in score and quality, as a dissertation for the degree of Doctor of Philosophy.



Jian Li
Assistant Professor of Electrical and
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in score and quality, as a dissertation for the degree of Doctor of Philosophy.



Joseph N. Wilson
Assistant Professor of Computer and
Information Science and Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

August 1996



Winfred M. Phillips
Dean, College of Engineering

Karen A. Holbrook
Dean, Graduate School